signals

MDPI

Article

An Empirical Study on Ensemble of Segmentation Approaches

Loris Nanni [1,*], Alessandra Lumini [2], Andrea Loreggia [3], Alberto Formaggio [1] and Daniela Cuza [1]

[1] Department of Information Engineering (DEI), University of Padua, Viale Gradenigo 6, 35131 Padova, Italy; alberto.formaggio@studenti.unipd.it (A.F.); daniela.cuza@studenti.unipd.it (D.C.)
[2] Department of Computer Science and Engineering (DISI), Università di Bologna, Via dell'Università 50, 47521 Cesena, Italy; alessandra.lumini@unibo.it
[3] Department of Information Engineering (DII), Università di Brescia, Via Branze 38, 25123 Brescia, Italy; andrea.loreggia@unibs.it
* Correspondence: loris.nanni@unipd.it

Abstract: Recognizing objects in images requires complex skills that involve knowledge about the context and the ability to identify the borders of the objects. In computer vision, this task is called semantic segmentation and it pertains to the classification of each pixel in an image. The task is of main importance in many real-life scenarios: in autonomous vehicles, it allows the identification of objects surrounding the vehicle; in medical diagnosis, it improves the ability of early detecting of dangerous pathologies and thus mitigates the risk of serious consequences. In this work, we propose a new ensemble method able to solve the semantic segmentation task. The model is based on convolutional neural networks (CNNs) and transformers. An ensemble uses many different models whose predictions are aggregated to form the output of the ensemble system. The performance and quality of the ensemble prediction are strongly connected with some factors; one of the most important is the diversity among individual models. In our approach, this is enforced by adopting different loss functions and testing different data augmentations. We developed the proposed method by combining DeepLabV3+, HarDNet-MSEG, and Pyramid Vision Transformers. The developed solution was then assessed through an extensive empirical evaluation in five different scenarios: polyp detection, skin detection, leukocytes recognition, environmental microorganism detection, and butterfly recognition. The model provides state-of-the-art results.

Keywords: machine learning; transfer learning; deep learning; ensembles; segmentation; transformers

Citation: Nanni, L.; Lumini, A.; Loreggia, A.; Formaggio, A.; Cuza, D. An Empirical Study on Ensemble of Segmentation Approaches. *Signals* **2022**, *3*, 341–358. https://doi.org/10.3390/signals3020022

Academic Editors: Yinsheng Chen, Aili Wang and Haibin Wu

Received: 21 April 2022
Accepted: 27 May 2022
Published: 1 June 2022

1. Introduction

Being able to recognize objects in images has been, for a long time, a prerogative of human beings. It has taken over 14 years to reach the level of an untrained human in the challenge of Imagenet. Things become more complex when the task requires not only to recognize the object in an image but also to identify its boundaries. This task is called semantic segmentation, and in machine learning this entails the classification of each pixel in an image. Due to the improvements of performance related to the adoption of machine learning models, this task is applied to many real-life scenarios [1,2]: in clinical practice, it can be used to identify polyps; similarly, in skin and blood analysis the identification of objects may help to visually bind the presence of different types of diseases. In addition, it can be used in autonomous vehicles, to identify objects surrounding the vehicle, in environmental microorganisms' classification, and many others.

The standard approach is to train a system composed of two modules: an encoder, and a decoder. The first module learns a low-dimensional representation of the input that describes semantics in the image. The second module learns to build the original input based on this low-dimensional feature vector. This has been the approach adopted by U-Net [3], one of the first systems developed for semantic segmentation. Autoencoders [4] were also employed to resolve the task due to their ability to learn semantics' low-level

representations of an image through the encoder module and the ability of reconstructing the original input from this reduced representation. Autoencoders' performance and results are the reasons why many researchers and practitioners from the computer vision area have adopted them.

The performance of autoencoders, as well as the ones of other classifier technologies, are strongly affected by architecture configuration, and other configurations often referred to as hyper-parameters tuning. That consists in finding the best values of some attributes of the model. This is a context-specific task that requires domain knowledge as well as expertise with the adopted machine learning techniques, resulting in big efforts and time consumption. The well-known no-free lunch theorem for machine learning highlights that a single model that works well on all the datasets cannot exist. Based on this evidence, another approach consists of adopting sets of classifiers, often shallow or weak, whose predictions are aggregated to form the output of the system. These frameworks are called ensemble methods. In an ensemble, individual classifiers are trained on the same dataset, in such a way, each model should generalize differently in the training space. Ensembles provide state-of-the-art results in many domains, but it is important to ensure some properties. One of them is to enforce some kind of diversity in the set of classifiers.

In this work, we propose a novel ensemble method for semantic segmentation. Our model is based on convolutional neural networks (CNNs) and transformers. Diversity among individual classifiers is enforced by adopting different loss functions and testing different data augmentations.

The model has been developed by combining DeepLabV3+ [5], HarDNet-MSEG [6], and Pyramid Vision Transformers [7]. The developed solution is then assessed through an extensive empirical evaluation that compares our proposal with state-of-the-art solutions highlighting promising results often better than best approaches.

The evaluation has been carried out on five different real-world scenarios, namely polyp segmentation, skin segmentation, butterfly recognition, environmental microorganisms' classification, and leukocyte detection.

Due to improvements of the discipline, machine learning techniques are used and applied in many different areas, for instance in medical diagnosis or in biology. Convolutional neural networks (CNNs) and other classic predictors are adopted for assisting researchers and practitioners in better identifying objects in images. This is the case for an instance of skin segmentation or butterfly identification. A drawback of this technology is based on the fact that a huge amount of data is needed to train these systems, but labeled data are a scarce resource in many domains. This is one of the reasons why big efforts are spent building and publishing datasets in specific areas; an example is Kvasir-SEG [8], a recent dataset that contains polyp images annotated at the pixel level by a group of experts.

A novel architecture came from the scope of natural language processing (NLP), where researchers study how to comprehend the semantic of texts with the purpose of automating tasks such as summarization or translation. This new model called the transformer is designed with a self-attention mechanism that enables the system to focus on a specific part of the input. Transformers have also been applied to computer vision tasks, gaining performances comparable or even better than CNNs. Once again, the main drawback of these models resides in the high demand of data useful to train a stable and performing system. TransFuse [9] and UACANet [10] are two recent approaches in the medical domain that combine different techniques: the first is a combination of CNN kernels and transformers; while the second blends U-Net and a parallel axial attention autoencoder. No matter the architecture, the aim is to capture information at both local and global levels.

As previously noticed, semantic segmentation becomes of main importance in many contexts. Autonomous vehicles, for instance, use semantic segmentation to identify objects in the environment surrounding the vehicle in order to make safe decisions [11]. In clinical practice, it helps in reducing the exposure to serious risks by detecting pathologies in their early stages, such as polyps detection that may prevent the evolution of colorectal cancers [2]. Similarly, in skin detection, deep learning methods are employed in various

areas, spanning from face detection to hand gesture recognition [12]. In this context, deep approaches have faced some difficulties, such as the clutter in the background that hinders the reliable detection of hand gestures in real-world environments.

CNNs have shown their appeal also to this context; two examples are the works from Roy et al. [13] and Arsalan et al. [14]. In the former work, the authors suggest using a CNN based on skin detection techniques to enhance the hand detector output. The latter instead introduces a residual skip connections (OR-Skip-Net) CNN that decreases the computational effort of the network and at the same time tackles demanding skin segmentation tasks; the goal is achieved by moving data to the last layer of the network directly from the initial layer. CNNs are also employed for automatically translating sign language [15].

A comparative analysis is reported in [12] through an extensive empirical evaluation of several leading technologies on a set of skin detection benchmarks.

Recently, deep learning was also used for the automatic recognition and classification of leukocytes [16]. This practice helps medical practitioners to diagnose various blood-related diseases. This can be done in many different ways: practitioners can analyze the percentages of leukocytes with a histogram-based technique or with iterative algorithms (such as GrabCut [17]) that segment white blood cells.

Contribution: this paper proposes a new ensemble method based on DeepLabV3+, HarDNet-MSEG, and Pyramid Vision Transformers' backbones. The proposal is intended to deal with semantic segmentation. In this work, diversity among individual classifiers in the ensemble is enforced by adopting different loss functions and testing different data augmentation approaches. We tested the proposed method on five different scenarios and compared the results with existing frameworks. The empirical evaluation highlights our results that are close to or even better than the state-of-the-art level.

2. Materials and Methods

In this section, we will provide all the techniques and approaches used to generate our ensemble. In particular, we will report the mathematical formalization of the loss functions adopted to design the networks.

2.1. Deep Learning for Semantic Image Segmentation

In the literature, different deep learning models are proposed to address semantic segmentation problems.

Semantic segmentation intends to identify objects in an image, with relative boundaries. The main purpose is therefore to assign a class at the pixel level; a task achieved thanks to FCNs (fully convolutional networks). FCNs have very high performance, and unlike CNNs they use a fully convolutional last layer instead of a fully connected one [18].

In order to obtain deconvolutional networks, such as U-Net, FCNs and autoencoders are combined together.

U-Net represents the first attempt to use autoencoders in an image segmentation task. Through the autoencoder, it is possible to downsample the input and simultaneously increment the number of features used to describe the input space. We can find another symbolic example in SegNet [19]: here, the max pool indices of the relative encoder level feed the decoders, while VGG is used for encoding. This allows it to reduce memory usage by having also a more promising segmentation.

DeepLab [20] consists of a series of autoencoder models introduced by Google, and has shown excellent results also in semantic segmentation applications [21]. These are some of the main features introduced to guarantee good performance:

1. A dilated convolution reduces the effects of pooling and stride, thereby greatly increasing the resolution.
2. Through an Atrous Spatial Pyramid Pooling, information is obtained at various scales.
3. A union of CNNs and probabilistic graphic models make it possible to detect the boundaries of objects.

We find in DeelLabV3, two most important innovations: a 1×1 convolution in Atrous Spatial Pyramid Pooling and a batch normalization; a set of modules placed in parallel and in cascade for convolutional dilation. DeepLabV3+ [5], an expansion of the family developed by Google, is adopted in this work. This expansion includes, among the most important features, a decoder with depth-wise convolutions and point-wise convolutions. The depth-wise works in the same location but with various channels, while the point-wise uses the same channel in various locations. In order to obtain different designs for a framework, we can consider other characteristics of the structure of a model, in fact the architecture model itself that is used is only one choice.

In this paper, we will investigate ResNet101 [22], a very famous CNN that acquires a residual function by referring to the block input (we recommend [23] for an exhaustive list of CNN structures). We adopt the ResNet101 network, pre-trained on the VOC segmentation dataset and then modulated through the parameters suggested, (https://github.com/matlab-deep-learning/pretrained-deeplabv3plus, (accessed on 20 April 2022)). These parameters have not been modified in order to prevent overfitting; thus, they are the same in all the tested datasets:

- initial learning rate = 0.01;
- number of epoch = 10 (using the simple data augmentation approach) or 15 (the latter more complex data augmentation approach due to the slower convergence using this larger augmented training set);
- momentum = 0.9;
- L2Regularization = 0.005;
- Learning Rate Drop Period = 5;
- Learning Rate Drop Factor = 0.2;
- Shuffle training images every-epoch;
- Optimizer = SGD (stochastic gradient descent).

Firstly, we propose an ensemble of the DeepLabV3+ models obtained by applying various loss and data augmentation methods, and then we combine the ensemble with HarDNet-MSEG [6], and Pyramid Vision Transformers (PVT) [7]. The HarD-Net-MSEG (harmonic densely connected network), a model influenced by densely connected networks, allows the reduction of memory consumption in this way: it decreases most of the connection layers at the DenseNet level, in order to reduce the costs of concatenation. In addition, the input/output channel ratio is equalized thanks to the increase in the channel width of the layers (consequently to the increase in its connections).

The PVT is a pure convolution-free transformer network which aims to acquire a high-resolution representation starting from a fine-grained input. The computational cost of the model is decreased by a progressive pyramidal shrinkage, accompanied by the depth of the model. A spatial-reduction attention (SRA) layer is introduced to an additional reduction of the computational complexity of the system.

In this work, both HarD-Net-MSEG and PVT have been trained using the same options in all the problems: batch size 15; number of epochs 100; initial learning rate 0.0001; decay rate = 0.1; decay epoch = 50.

2.2. Loss Functions

Some loss functions tested for the design of the networks' ensemble will be presented in this section, including the loss functions suggested in [24] (Section 2.2.1) and the new ones here tested/proposed (Sections 2.2.2–2.2.7).

Different loss functions influence the training phase and the performance of the model. In semantic segmentation tasks, pixel-wise-cross-entropy is one of the most widespread and adopted loss functions; this operates at the pixel level, verifying whether the predicted label of a given pixel coincides with the ground-truth. One of the main problems with this approach is the critical situation in which the dataset is unbalanced with respect to the labels, but it can be solved through the use of counterweights. The dice loss function [25]

aims to verify the overlap between the predicted segmented images and the ground-truth. This approach, that we have also used in this work, is widespread in semantic segmentation.

An exhaustive overview of image segmentation and loss functions is available in a recent survey [25].

2.2.1. Previously Tested Loss Functions

Our set of loss function includes the following widely adopted metric functions (the interested reader can refer to [24] for a more detailed description):

- Generalized Dice Loss [26], $L_{GD}(Y,T)$ is a multiclass variant of the dice loss.
- Tversky Loss [27], $L_T(Y,T)$ is a weighed version of the Twersky index designed to deal with unbalanced classes, i.e., the phenomenon whereby one class prevails over another.
- Focal Tversky Loss [28], $L_{FT}(Y,T)$ it is a variant of the Twersky loss, where a modulating factor ($\gamma = 4/3$) is used to ensure that the model focuses on hard samples instead of properly classified examples.
- Focal Generalized Dice Loss [29], $L_{FGD}(Y,T)$ is the focal version of the generalized dice loss.
- 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 curve of loss.
- Log-Cosh Focal Tversky Loss $L_{lcFT}(Y,T)$ is based on the same idea of smoothing, here applied to the focal Tversky loss.
- SSIM Loss [30], $L_S(Y,T)$, is obtained from the structural similarity (SSIM) index [31], usually adopted to evaluate the quality of an image.
- MS-SIM Loss $L_{MS}(Y,T)$ is a variant of $L_S(Y,T)$ defined using the multiscale structural similarity (MS-SSIM) index.
- Some combined losses:
  - $Comb_1\ (Y,T) = L_{FGD}(Y,T) + L_{FT}(Y,T)$
  - $Comb_2(Y,T) = L_{lcGD}(Y,T) + L_{FGD}(Y,T) + L_{lcFT}(Y,T)$
  - $Comb_3\ (Y,T) = L_S(Y,T) + L_{GD}(Y,T)$
  - $Comb_4\ (Y,T) = L_{MS}(Y,T) + L_{FGD}(Y,T)$

2.2.2. Cross-Entropy

The cross-entropy (CE) loss function provides us a measure of the difference between two probability distributions. The goal is to minimize this difference and, in doing so, it has no bias between small or large regions.

This could be an issue when dealing with imbalanced datasets. Hence, the weighted cross-entropy loss was introduced and it resulted in well-balanced classifiers for imbalanced scenarios [32].

The formula for weighted binary cross-entropy is presented in (1). In this equation, T refers to the ground truth label image, while T_{ik} is the true value for the pixel i and it can be equal to either 0 or 1. It is equal to 0 if the pixel i belongs to the class k, 0 otherwise.

P is the prediction for the output image and P_{ik} is the probability of the i-th pixel to belong to the k-th class obtained by using the sigmoid activation function. For P, we used the softmax activation function which returns probabilities.

w_{ik} is the weight given to the i-th pixel of the image for the class k. These weights were calculated by using an average pooling over the mask with a kernel 31×31 and a stride of 1 in order to consider also nonmaximal activations.

$$L_{WBCE} = -\sum_{k=1}^{K}\sum_{i=1}^{N} w_{ik} \times T_{ik} \times \log(P_{ik}) \quad (1)$$

where K is the number of classes and N the number of pixels.

2.2.3. Weighted Intersection over Union

Another well-known loss function is intersection over union (IoU) loss, which was introduced for the first time in [33]. The original formula was:

$$\text{IoU} = \frac{|P \cap T|}{|P \cup T|} \tag{2}$$

As mentioned earlier, T is the truth label image and P is the prediction for the output image.

Unfortunately, the set symbols for Intersection and Union are not differentiable because P and T have to be either 0 s or 1 s. This is not true for P, so the formula was then approximated with the following:

$$\text{IoU}' = \frac{|P \times T|}{|P + T - P \times T|} \tag{3}$$

where $P \times T$ is the element-wise multiplication of T and P. For what concerns the denominator, we subtract the "intersection" between P and T because we do not want to consider the intersection twice.

Once the set operators have been converted to arithmetic ones, the formula is differentiable, and it is possible to evaluate the gradient.

However, IoUis an evaluation metric used for evaluating the goodness of the prediction. Hence, a value of 1 is equivalent to a perfect prediction. For this reason, the loss function will be:

$$L_{\text{IoU}} = 1 - \text{IoU}' \tag{4}$$

Unfortunately, this function has to face the same problem of CE in inferring the label of the boundary of each object; therefore, as suggested in [34], we use the weighted intersection over union () wIoU, instead of the standard IoU.

The formula of the weighted intersection over union loss is:

$$L_{\text{wIoU}} = 1 - \frac{|w * P * T|}{|w * (P + T) - w * P * T|} = 1 - \frac{\sum_{i=1}^{N} w_{ik} * \sum_{k=1}^{K} T_{ik} * Y_{ik} + 1}{\sum_{i=1}^{N} \sum_{k=1}^{K} w_{ik}(T_{ik} + Y_{ik} - T_{ik} * Y_{ik}) + 1} \tag{5}$$

where N is the number of pixels and K is the number of classes. The weights w_{ik} are calculated as aforementioned. T_{ik} and Y_{ik} are, respectively, the ground truth value and the prediction value for the pixel i belonging to the class k. We added 1 to both the numerator and the denominator in order to prevent the undefined division $\frac{0}{0}$.

2.2.4. Structure Loss

Based on the intuition in [6], weighted intersection over union and weighted binary crossed-entropy are considered together.

$$L'_{STR} = L_{\text{wIoU}} + L_{wbce} \tag{6}$$

Instead of applying an avgpool over the mask, we do so over the predictions. We have done this to improve the diversity in the deep learning network.

Then, we want to give more importance to the binary crossed-entropy loss, so we use a weight of 2 for that one, and this leads to the following formulation:

$$L_{STR} = L_{\text{wIoU}} + 2L_{wbce} \tag{7}$$

2.2.5. BoundExpStructure

We combine structure loss, boundary loss and exponential logarithmic loss. This is done to have better performances on the small structures of a highly imbalanced dataset and, at the same time, have better identification of the boundaries of the image.

$$L_{BoundExpS} = L_{Bound} + L_{Exp} + L_{Str} \tag{8}$$

2.2.6. Boundary Enhancement Loss

The boundary enhancement loss is a loss proposed in [35] which explicitly focuses on the boundary areas during training.

This loss has very good performances as it does not require neither any pre- or post-processing of the image nor a particular net in order to work.

The Laplacian filter $\mathcal{L}(\cdot)$ is the milestone of this loss as it generates strong responses around the boundaries and zero everywhere else. When applying the Laplacian filter to a mask S, we obtain:

$$\mathcal{L}(\mathrm{x}, \mathrm{y}) = \frac{\partial^2 S}{\partial x^2} + \frac{\partial^2 S}{\partial y^2} \tag{9}$$

The positive aspect about using the Laplacian filter, is that it can be achieved quite easily by a series of convolution operations. So, the idea is to evaluate the difference between the filtered output of ground-truth labels and the filtered output of the predictions.

The boundary enhancement loss, as proposed in [35], is:

$$L_{BE} = ||\mathcal{L}(T) - \mathcal{L}(Y)||_2 = \left|\left|\frac{\partial^2 (T-Y)}{\partial x^2} + \frac{\partial^2 (T-Y)}{\partial y^2}\right|\right|_2 \tag{10}$$

where $||\ ||_2$ is the l_2 norm. This can be easily achieved as already described in the original paper [35].

Based on the idea of the same paper, we will be using dice loss and boundary enhancement loss together, weighted appropriately, and the structure loss:

$$L_{DiceBES} = \lambda_1 L_{Dice} + \lambda_2 L_{BE} + L_{Str} \tag{11}$$

The best results are achieved by using $\lambda_1 = 1$ and $\lambda_2 = 0.01$.

2.2.7. Contour-Aware Loss

Contour-aware loss was proposed for the first time in [36]. It consists in a weighted binary cross-entropy loss where the weights are obtained with the aim of giving more importance to the borders of the image.

In the loss is employed a morphological gradient edge detector. Basically, the difference between the dilated and the eroded label map is evaluated. Then, for smoothing purposes, the Gaussian blur is applied. This spatial weight map can be formulated as:

$$M^C = Gauss(K \cdot (dilate(T) - erode(T))) + 1 \tag{12}$$

here $dilate(T)$ and $erode(T)$ are dilation and erosion operations with a 5×5 kernel. K is a hyperparameter for assigning the high value to contour pixels, which is set to 5 empirically. 1 is the matrix with 1 in every position.

We can compute the contour-aware loss as:

$$L_C = -\sum_{i=1}^{N} M_i^C \times (T_i \times \log(Y_i) + (1 - T_i) \times \log(1 - Y_i)) \tag{13}$$

Finally, we compute the new loss we are going to use:

$$L_{CS} = L_C + L_{Str} \tag{14}$$

2.3. Data Augmentation

The training phase of a classifier and the resulting performance of the system are strongly influenced by the size of the dataset. This is true also for an ensemble method. Thus, to increase the amount of data that can be used to train the system, several techniques may be applied to the original dataset. In the next paragraphs, we shall describe the different techniques adopted with the purpose of data augmentation. We employ these techniques on the training set, both on the input samples and their mask. We leave the test set unchanged.

Two different data augmentation approaches are tested:

- DA1, base data augmentation consisting in horizontal and vertical flip, 90° rotation.
- DA2, the following operations are performed:
  1. The image is displaced to the right or the left.
  2. The image is displaced up or down.
  3. The image is rotated by an angle randomly selected from the range [0° 180°].
  4. Horizontal or vertical shear by using the Matlab function *randomAffine2d()*.
  5. Horizontal or vertical flip.
  6. Change in the brightness levels by adding the same value (random value between 25 and 50) to each RGB channel.
  7. Change in the brightness levels by adding different values (random value between 25 and 50) to each RGB channel.
  8. Add speckle noise, it adds multiplicative noise to the image I, adding a value n × I, where n is uniformly distributed random noise with mean 0 and variance 0.05.
  9. Application of the technique "Contrast and Motion Blur", described below.
  10. Application of the technique "Shadows", described below.
  11. Application of the technique "Color Mapping", described below.

Some artificial images (DA2 approach) contain only background pixels; therefore, to discard them we simply delete all the images where there are less than 10 pixels that belong to the foreground class.

2.3.1. Shadows

New image samples can be created by creating shadows in the original set of images. Shadows may be created randomly to the left or to the right of the original image. We use the following criteria to decide the intensity of the shadow (*direction* = 1: right; *direction* = 0: left):

$$y = \begin{cases} \min\left\{0.2 + 0.8\sqrt{\frac{x}{0.5}},\ 1\right\} & direction = 1 \\ \min\left\{0.2 + 0.8\sqrt{\frac{1-x}{0.5}},\ 1\right\} & direction = 0 \end{cases} \tag{15}$$

2.3.2. Contrast and Motion Blur

Another technique for data augmentation that allows one to derive new samples from an original dataset is based on the combination of contrast and motion blur. The first one increases or decreases the original contrast, and the second one simulates the movement of the camera taking the image. We developed two different contrast functions, and each time we choose one of them at random.

The first function is defined as follows:

$$y = \frac{\left(x - \frac{1}{2}\right)\sqrt{1 - \frac{k}{4}}}{\sqrt{1 - k\left(x - \frac{1}{2}\right)^2}} + 0.5,\ k \leq 4 \tag{16}$$

The parameter k controls the contrast. Specifically: T\hte contrast increases when $k < 0$; it is decreased when $0 < k \leq 4$; the image is unchanged when $k = 0$.

The value of the parameter is drawn at random in the following range:

- $\mathcal{U}(2.8,\ 3.8) \rightarrow$ Hard decrease in contrast;
- $\mathcal{U}(1.5,\ 2.5) \rightarrow$ Soft decrease in contrast;
- $\mathcal{U}(-2, -1) \rightarrow$ Soft increase in contrast;
- $\mathcal{U}(-5, -3) \rightarrow$ Hard increase in contrast.

The second function is defined as follows:

$$y = \begin{cases} \frac{1}{2}\left(\frac{x}{0.5}\right)\alpha \ 0 \leq x < \frac{1}{2} \\ 1 - \frac{1}{2}\left(\frac{1-x}{0.5}\right)\alpha \ \frac{1}{2} \leq x \leq 1 \end{cases} \tag{17}$$

The parameter α controls the contrast. In particular, the contrast is increased when $\alpha > 1$; it is decreased when $0 < \alpha < 1$; if $\alpha = 1$, then the image is left unchanged.

The parameter is chosen randomly from four possible ranges:

- $\mathcal{U}(0.25,\ 0.5) \rightarrow$ Hard decrease in contrast;
- $\mathcal{U}(0.6,\ 0.9) \rightarrow$ Soft decrease in contrast;
- $\mathcal{U}(1.2,\ 1.7) \rightarrow$ Soft increase in contrast;
- $\mathcal{U}(1.8,\ 2.3) \rightarrow$ Hard increase in contrast;

The blurring effect that mimics the movement of the camera is applied right after the contrast adjustment. We use the MATLAB function *fspecial('motion', len, theta)*.

2.3.3. Color Mapping

Changing the color map of the image produces a new image. In particular, it is possible to map the color of an image to that of another image. We generated a pair of images by coupling any image in the original training set with another randomly selected image in the same set. We adopt the stain normalization toolbox (the toolbox is authored by Nicholas Trahearn and Adnan Khan and available online at https://warwick.ac.uk/fac/cross_fac/tia/software/sntoolbox/, (accessed on 20 April 2022)), which provides this functionality in three different versions:

- RGB Histogram Specification;
- Reinhard;
- Macenko.

3. Results

We run an extensive empirical evaluation with the aim of evaluating the performance of our ensemble. We comprehend a comparison with several state-of-the-art models for a more exhaustive evaluation of our system. The empirical evaluation is carried out on five real-world scenarios: polyp segmentation, skin segmentation, leukocyte identification, butterfly, and microorganism identification.

3.1. Datasets and Testing Protocols

Some examples for images and masks, from each of the five datasets, are displayed in Figure 1. It is clear that they are very different segmentation problems.

3.1.1. Polyp Segmentation (POLYP)

Polyp segmentation from colonoscopy is a challenging task requiring a two-class classification between the low contrast colon background and polyp foreground pixels.

We present experimental results according to a very popular benchmark [6] available on GitHub (https://github.com/james128333/HarDNet-MSEG, (accessed on 20 April 2022)), and including five datasets for polyp segmentation (i.e., Kvasir [37], ColonDB [38], CVC-T [39] and ETIS [40], ClinicalDB [41]): the training set is made by 1450 images (from the most large dataset, i.e., 900 images from Kvasir and 550 images from ClinicalDB); the other are for the test sets (100 images from Kvasir, 380 from ColonDB, 60 from CVC-T,

196 from ETIS, and 62 from ClinicalDB), as is usually done in the literature. In these datasets, we use resized training images of size 352 × 352.

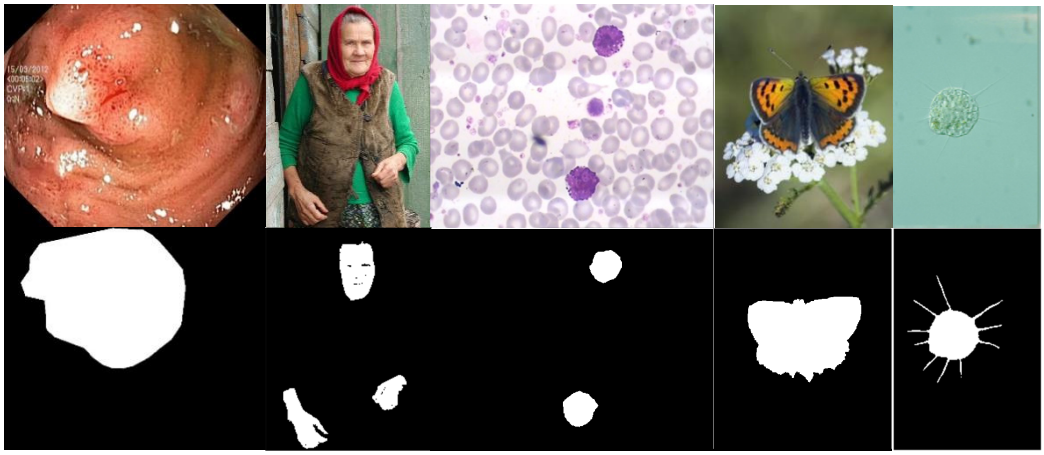

Figure 1. Samples from polyp segmentation, skin segmentation, leukocyte identification, butterfly, and microorganism identification (images and masks).

3.1.2. Skin Segmentation (SKIN)

The segmentation task in skin detection is recognizing the image portions that represent "skin" and "no skin": as a result, it is a binary classification problem. We use the framework proposed in [12] in this paper, which includes a small training set of 2000 images from the ECU dataset [42] and ten testing datasets very different to each other, see Table 1. According to the original testing protocol [12], dice (i.e., F1-score) is calculated at the pixel level and not at the image level and average for the whole dataset. In these datasets, we use resized training images of size 352 × 352.

Table 1. Test skin datasets. ECU is split: 2000 images for training and 2000 for tests.

Short Name	Name	#Samples	Ref.
Prat	Pratheepan	78	[43]
MCG	MCG-skin	1000	[44]
UC	UChile DB-skin	103	[45]
CMQ	Compaq	4675	[46]
SFA	SFA	1118	[47]
HGR	Hand gesture recognition	1558	[48]
Sch	Schmugge dataset	845	[49]
VMD	Human activity recognition	285	[50]
ECU	ECU face and skin detection	2000	[42]
VT	VT-AAST	66	[51]

3.1.3. 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 experiment, we use the freely available (http://users.cecs.anu.edu.au/~hrezatofighi/Data/Leukocyte%20Data.htm, (accessed on 20 April 2022)) LISC database [16], 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 type of leukocytes. In this work, we do not perform classification; therefore, we consider only the segmentation performance. The testing protocol, as suggested by the authors of the dataset, is a 10-fold cross validation: dice results are calculated at the image level and averaged for each fold and then on the 10 folds. In this dataset, we use resized images of size 513 × 513.

3.1.4. Butterfly Identification (BFLY)

As already completed in the literature, for butterfly identification we adopt the public dataset (http://www.josiahwang.com/dataset/leedsbutterfly/, (accessed on 20 April 2022)) [52]. For a fair comparison with the previous results, we have used the same testing protocol proposed by the authors of the dataset; that is, a four-fold cross validation, where each fold includes 208 test images and 624 training images. In this dataset, we use resized images of size 513×513.

3.1.5. Microorganism Identification (EMICRO)

EMicro [53] is a public dataset (https://figshare.com/articles/dataset/EMDS-6/17125025/1, (accessed on 20 April 2022)) of Environmental Microorganism Image Dataset Sixth Version (EMDS-6). It is composed of 840 images: following the original paper, we split the dataset and the 37.5% of the images belong to the test set.

In this dataset, we use resized images of size 513×513.

3.2. Metrics

The system has been evaluated using two standard metrics: dice score and intersection over union (IoU). In the following formula, TP, TN, FP, and FN correspond to the true positives, true negatives, false positives, and false negatives, respectively. A is the predicted mask (TP + FP) and B is the ground truth map (TP + FN).

Dice score (which is equivalent to F1score in binary classification tasks) is a weighted average of precision and recall. Formally, it is defined as:

$$F1score = Dice = \frac{|A \cap B|}{|A| + |B|} = \frac{2 \cdot \mathrm{TP}}{2 \cdot \mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{18}$$

Intersection over union (IoU) defines the shared area between two masks, divided by the area of the union between the two maps. Formally, it is defined as:

$$\mathrm{IoU} = \frac{|A \cap B|}{|A \cup B|} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{19}$$

In the experiments, image size has been modified due to the input size of the models. In these cases, we always change back to the original dimension the predicted masks.

3.3. Experiments

3.3.1. Baseline Ensembles

As the aim of this paper is to study approaches to increase the diversity of ensembles, we report in Table 2 the performance of some baseline classifiers and ensembles based on different network architectures (all combined with the data augmentation DA1, see Section 2.3). The tests reported in Section 3.3.1 are all based only on the dice loss; moreover, for the sake of space, for the polyp and skin datasets we report only the average performance value among the set of datasets. Each ensemble is made up of N models ($N = 1$ denotes a stand-alone model) which differ only for the randomization in the training process:

- *RN18* is a stand-alone DeepLabV3+ segmentator with backbone Resnet18 (pretrained in ImageNet);
- *ERN18*(N) is an ensemble of N *RN18* networks (pretrained in ImageNet);
- *RN50* is a stand-alone DeepLabV3+ segmentator with backbone Resnet50 (pretrained in ImageNet);
- *ERN50*(N) is an ensemble of N RN50 networks;
- *RN101* is a stand-alone DeepLabV3+ segmentator with backbone Resnet101 (pretrained as detailed in the above Section 2.1);
- *ERN101*(N) is an ensemble of N *RN101* networks.

Table 2. Performance (dice) of the proposed ensembles in the five benchmark datasets; the last column *AVG* reports the average performance. Best performance is marked in bold font.

	POLYP	**SKIN**	**LEUKO**	**BFLY**	**EMICRO**	***AVG***
RN18	0.806	0.865	0.897	0.960	0.908	0.887
RN50	0.802	0.871	0.895	0.968	0.909	0.889
RN101	0.808	0.871	0.915	0.976	0.918	0.898
ERN18(10)	0.821	0.866	0.913	0.963	0.913	0.895
ERN50(10)	0.807	0.872	0.897	0.969	0.918	0.893
ERN101(10)	**0.834**	**0.878**	**0.925**	**0.978**	**0.919**	**0.907**

The results in Table 2 show that, although the overall performance increases when switching from the stand-alone version to an ensemble, the improvement is not as high as one might expect, indicating that the individual approaches are quite stable. Maybe this result is related to the architecture of the DeepLabV3+ network: its internal modules apply atrous convolutions in cascade or in parallel to capture a multi-scale context by adopting several atrous rates. This solution, which has been designed to solve the problem of segmenting objects at multiple scales, also mimics an ensemble approach thanks to the fusion of activations taken at different levels of the encoder, making the resulting segmentation quite stable. The best method is to use ResNet101 as a backbone.

3.3.2. Ablation Studies

The first ablation study is related to the evaluation of different loss functions to increase the diversity of the models and improve the performance of the ensemble. In Table 3, the performance of RN101, with the different loss functions here proposed, is reported and compared with the dice loss as the baseline and DA1 as the data augmentation method. For the sake of space, for the polyp and skin datasets we report only the average performance value among the test sets. Then, the stand-alone networks are fused, always using the sum rule, in some ensembles:

- *ELoss101*(*10*) is an ensemble, combined by the sum rule, of 10 RN101, each coupled with data augmentation DA1 and a given loss function. The final fusion is given by: $2 \times L_{GD} + 2 \times L_T + 2 \times Comb1 + 2 \times Comb2 + 2 \times Comb3$; where with $2 \times L_{GD}$ we mean two RN101 trained using L_{GD} loss function;
- *ELossMix*(*10*) is an ensemble similar to the previous one, but here data augmentation is used to increase diversity: the networks coupled with the loss used in *ELoss101*(*10*) (L_{GD}, L_T, Comb1, Comb2, Comb3) are trained one time using DA1 and another time using DA2 (i.e., 5 networks each trained two times, so we have an ensemble of 10 networks);
- *ELossMix2*(*10*), it is similar to the previous ensemble, but using $L_{DiceBES}$ instead of L_T
- *ELossLarge*(*10*) is an ensemble of 10 networks: the networks trained using (L_{GD}, L_T, Comb1, Comb2, Comb3) use DA2 as the augmented training set, while the networks trained considering the new loss functions tested in this work (L_{STR}, $L_{BoundExpS}$, $L_{DiceBES}$, L_{MS}, L_{CS}) are coupled with DA1.

The results reported in Table 3 show that the proposed new loss functions gain performance similar to dice and can be considered a useful starting point for the design of an ensemble. In fact, the good performances of *ELoss101* and *ELossLarge* with respect to *ERN101*(*10*) prove that the including networks trained by different loss functions are useful for the creation of an ensemble: this observation is even more evident considering that it is validated on very different problems.

We should not overlook the value of altering the training set, in this case through the use of different data augmentation: this strategy seems to be the winner when combined with different loss functions (*ELossMix*). *ELossMix2* seems to be more stable than *ELossMix*. Finally, we can notice that *ELossMix* and *ELossLarge* obtain a similar performance.

Table 3. Performance (dice) of some stand-alone methods and ensembles in the five benchmark datasets when varying the loss function; the last column *AVG* reports the average performance. Best performance is marked in bold font.

	LOSS	POLYP	SKIN	LEUKO	BFLY	EMICRO	*AVG*
RN101	L_{GD}	0.808	0.871	0.915	0.976	0.918	0.898
RN101	L_{STR}	0.809	0.869	0.930	0.964	0.901	0.895
RN101	$L_{BoundExpS}$	0.803	0.874	0.928	0.978	0.901	0.897
RN101	$L_{DiceBES}$	0.819	0.869	0.922	0.969	0.904	0.897
RN101	Comb4	0.813	0.860	0.920	0.972	0.920	0.897
RN101	L_{CS}	0.823	0.873	0.917	0.967	0.911	0.898
ERN101(10)	L_{GD}	0.834	0.878	0.925	0.978	0.919	0.907
ELoss101(10)	Many loss	0.842	0.880	0.925	0.980	0.921	0.910
ELossMix(10)	Many loss	**0.851**	**0.883**	0.936	0.983	**0.924**	0.915
ELossMix2(10)	Many loss	**0.851**	**0.883**	0.943	**0.984**	**0.925**	**0.917**
ELossLarge(10)	Many loss	0.848	**0.883**	**0.944**	**0.984**	0.922	**0.916**

The second ablation study is related to the evaluation of different architectures. The performance of the above cited methods coupled with different data augmentation strategies are reported in Table 4. The names DA1 and DA2 refer to the strategies explained in Section 2, while DA1/2 denotes that the given ensemble is obtained by the fusion of networks based on DA1 and networks based on DA2. HardNet-MSEG is trained with two different optimizers (SGD denoted as H_S and Adam denoted as H_A), and the ensemble FH is the fusion of HarDNet-MSEG trained with different optimizers. PVT is trained using the AdamW optimizer (as suggested in the original paper where PVT has been proposed). The loss functions for both HarDNet-MSEG and PVT are the same as the original papers.

Table 4. Performance (dice) of some stand-alone methods and ensembles in the five benchmark datasets. Best performance is marked in bold font.

	DA	POLYP	SKIN	LEUKO	BFLY	EMICRO	*AVG*
ELossMix(10)	DA1/DA2	0.851	0.883	0.936	0.983	0.924	0.915
H_A	DA1	0.840	0.867	0.923	0.977	0.914	0.904
H_A	DA2	0.854	0.871	0.945	0.982	0.912	0.913
H_S	DA1	0.816	0.872	0.889	0.969	0.894	0.880
H_S	DA2	0.847	0.870	0.917	0.976	0.901	0.902
FH	DA1	0.859	0.879	0.913	0.980	0.915	0.909
FH(2)	DA1/2	0.862	0.885	0.934	0.982	0.916	0.916
PVT	DA1	0.854	0.878	0.954	0.975	0.920	0.916
PVT	DA2	0.855	0.879	0.954	0.984	0.919	0.918
PVT(2)	DA1/2	0.855	0.883	0.957	0.984	0.922	0.920
$FH(2) + 2 \times PVT(2)$	DA1/2	**0.875**	0.892	**0.955**	**0.985**	0.924	**0.926**
$ELossMix(10) + (10/4) \times FH(2) + (10/2) \times PVT(2)$	DA1/2	**0.875**	**0.893**	0.953	**0.985**	**0.926**	**0.926**
$ELossMix2(10) + (10/4) \times FH(2) + (10/2) \times PVT(2)$	DA1/2	**0.874**	**0.893**	0.954	**0.985**	**0.926**	**0.926**

Some further ensembles are reported in Table 4:

- *PVT*(2), sum rule between PVT combined with DA1 and PVT combined with DA2;
- *FH*(2), sum rule among two H_S (one combined with DA1, the latter with DA2) and two H_A (one combined with DA1, the latter with DA2);
- $FH(2) + 2 \times PVT(2)$, weighted sum rule between *PVT*(2) and *FH*(2), the weight of *PVT*(2) is assigned so that its importance in the ensemble is the same as *FH*(2) (notice that *FH*(2) consists of four networks while *PVT*(2) is built by only two networks);
- $ELossMix(10) + (10/4) \times FH(2) + (10/2) \times PVT(2)$, weighted sum rule among *ElossMix*(*10*), *FH*(2), and *PVT*(2); as in the previous ensemble, the weights are assigned so that each ensemble member has the same importance (notice that *ElossMix*(*10*) is the fusion by sum rule of 10 DeepLabV3+).

The results reported in Table 4 permit us to draw the following conclusions:

- *PVT*(2), *FH*(2), and *ElossMix*(10) obtain very similar performances, and only in Leuko the performance of *PVT*(2) is better than *FH*(2) and *ElossMix*(10);
- *PVT*(2) permits obtaining a very slight performance improvement with respect to stand-alone PVT, even the improvement of *FH*(2) with respect to the best stand-alone HardNet (i.e., H_A combined with DA2);
- Combining different architectures permits obtaining the best performance, and the best trade-off "complexity vs performance" is given by "*FH*(2) + 2 × *PVT*(2)".

3.3.3. Comparison with the Literature

For comparison with other approaches in the literature, the results of our best ensembles are reported in full in the different datasets of polyp segmentation (Table 5) and skin detection (Table 6). The following tests clearly show that *FH*(2) + 2 × *PVT*(2) obtains a state-of-the-art performance.

Table 5. Performance (dice and IoU) in the polyp segmentation problem. Best performance is marked in bold font.

Method	Kvasir		ClinicalDB		ColonDB		ETIS		CVC-T		*Average*	
	IoU	Dice	IoU	Dice	IoU	Dice	IoU	Dice	IoU	Dice	IoU	Dice
FH(2) + 2 × *PVT*(2)	**0.874**	**0.920**	0.894	0.937	**0.751**	**0.826**	**0.717**	**0.787**	0.842	0.904	**0.816**	**0.875**
Ensemble in [24]	0.871	0.917	0.886	0.931	0.697	0.769	0.663	0.740	0.829	0.901	0.790	0.852
HarDNet-MSEG [6]	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 [6])	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 [6])	0.611	0.723	0.607	0.700	0.347	0.469	0.217	0.297	0.329	0.467	0.422	0.531
U-Net++ (from [6])	0.743	0.821	0.729	0.794	0.41	0.483	0.344	0.401	0.624	0.707	0.570	0.641
U-Net (from [6])	0.746	0.818	0.755	0.823	0.444	0.512	0.335	0.398	0.627	0.710	0.581	0.652
SETR [21]	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 [54]	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 [9]	0.870	**0.920**	**0.897**	**0.942**	0.706	0.781	0.663	0.737	0.826	0.894	0.792	0.855
UACANet [10]	0.859	0.912	0.88	0.926	0.678	0.751	0.678	0.751	**0.849**	**0.910**	0.789	0.850
SANet [55]	0.847	0.904	0.859	0.916	0.670	0.753	0.654	0.750	0.815	0.888	0.769	0.842
MSNet [56]	0.862	0.907	0.879	0.921	0.678	0.755	0.664	0.719	0.807	0.869	0.778	0.834
PVT [7]	0.864	0.917	0.889	0.937	0.727	0.808	0.706	**0.787**	0.833	0.900	0.804	0.869
SwinE-Net [57]	0.870	**0.920**	0.892	0.938	0.725	0.804	0.687	0.758	0.842	0.906	0.803	0.865
AMNet [58]	0.865	0.912	0.888	0.936	0.690	0.762	0.679	0.756	-	-	-	-

Table 6. Performance (dice = F1-score) in the skin detection problem. Best performance is marked in bold font.

	DA	PRAT	MCG	UC	CMQ	SFA	HGR	SCH	VMD	ECU	VT	*AVG*
ERN101(1)	DA1	0.922	0.887	0.923	0.823	0.948	0.969	0.750	0.748	0.948	0.796	0.871
ERN101(10)	DA1	0.924	0.887	0.920	0.845	0.952	**0.971**	0.778	0.754	0.950	0.794	0.878
ELoss101(10)	DA1	0.926	0.892	0.923	0.844	**0.956**	**0.971**	0.777	0.751	0.953	0.807	0.880
ELossMix(10)	DA1/DA2	0.924	0.893	0.929	0.850	**0.956**	0.970	0.789	0.739	0.952	0.829	0.883
H_S	DA1	0.903	0.880	0.903	0.838	0.947	0.964	0.793	0.744	0.941	0.810	0.872
H_A	DA1	0.913	0.880	0.900	0.809	0.951	0.967	0.792	0.717	0.945	0.799	0.867
PVT	DA1	0.920	0.888	0.925	0.851	0.951	0.966	0.792	0.709	0.951	0.828	0.878
FH(2) + 2 × *PVT*(2)	DA1/DA2	**0.927**	**0.894**	**0.932**	**0.868**	0.954	**0.971**	0.797	**0.767**	**0.955**	**0.853**	**0.893**
Ensemble in [24]	DA1	0.926	0.888	0.916	0.842	0.955	**0.971**	**0.799**	0.764	0.952	0.820	0.883

In LEUKO, the authors of the dataset report an IoU of 0.842, and *FH*(2) + 2 × *PVT*(2) obtains a higher IoU of 0.916.

In EMicro, the authors of the dataset report a dice score of 0.884, and *FH*(2) + 2 × *PVT*(2) obtains a higher dice of 0.924.

In the BFLY, many approaches have been tested (see [59]), and the two best methods reported in the literature are:

(a) Ref. [59] that reports an IoU of 0.950;
(b) Ref. [60] that reports an IoU of 0.945.

Our suggested ensemble (i.e., $FH(2) + 2 \times PVT(2)$) strongly outperforms previous state-of-the-art methods obtaining an IoU of 0.970.

Clearly, the ensemble boosts the performance of the best stand-alone network (PVT combined with DA2).

The main cons of this approach are that the best ensemble is composed by six networks, and this means 6× RAM requirements and 6× inference time. Furthermore, the inference time is very low also using an ensemble, and with the current GPU architectures it is not a problem in many issues (obviously it could be a problem in some applications such as autonomous drive, but not in the segmentation problems faced in this paper).

E.g., a single HarDNet-MSEG runs at 86.7 images/second on a GeForce RTX 2080 Ti GPU.

We have performed a further experiment to select the optimal set of models to be included in the final ensemble. We have extracted a validation set to select the best set of networks: we have considered only the two problems including many test sets, i.e., polyp and skin segmentation. In the polyp problem, the Kvasir test set has been chosen as a validation set; in the skin application problem, the ECU test set has been used as validation set. We have used a sequential forward floating selection (SFFS) [61] for retaining the subset of networks that maximize the dice performance indicator in the validation set. The performance of both ensembles were lower than we expected and in both datasets our best approach (i.e., $FH(2) + 2 \times PVT(2)$) gained a higher performance. In both cases, we have faced an overfitting problem: the images in test sets are very different among each other; therefore, a larger validation set, and more comprehensive different variations that can occur to an image, are needed for a reliable network selection.

Finally, we perform some tests using the Q-statistic for further validation of our idea to build ensembles. Yule's Q-statistic [62] was conducted to demonstrate the relationship of diversity among the networks that belong to the ensemble. After calculation, the range of Q-statistic varies from -1 to 1. For statistically independent classifiers, the Q-statistic is equal to zero.

In Table 7, we report the average Q-statistic among the network of the proposed ensemble; clearly, *ELossMix* allows for a set of networks with greater diversity than *Eloss101* and *ERN101*. Moreover, $FH(2) + 2 \times PVT(2)$ is built by a set of quite different segmentators.

Table 7. Average Q-statistic.

Ensembles	Average Q-Statistic
ERN101(*10*)	0.975
ELOSS101(*10*)	0.952
ELOSSMIX(*10*)	0.921
$FH(2) + 2 \times PVT(2)$	0.925

4. Conclusions

In computer vision, we call semantic segmentation the task that involves the classification of each pixel in an image.

This is a very important task in several fields, e.g., in autonomous vehicles, it allows the identification of objects surrounding the vehicle; in medical diagnosis, it improves the ability of early detecting dangerous pathologies and thus mitigates the risk of serious consequences.

Here, we obtain a state-of-the-art performance proposing different ensembles of segmentation approaches. We have tested:

- Different loss functions;

- Different data augmentation approaches;
- Different network topologies, i.e., convolutional neural networks and transformer (namely DeepLabV3+, HarDNet-MSEG, and Pyramid Vision Transformers).

Finally, the ensemble is combined by the sum rule.

Our proposed ensemble has been tested, providing state-of-the-art results, in five benchmark datasets: polyp detection, skin detection, leukocytes recognition, environmental microorganism detection, and butterfly recognition.

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

All resources are available online at https://github.com/LorisNanni, (accessed on 20 April 2022).

Author Contributions: Conceptualization, L.N., D.C. and A.F.; methodology, L.N. and A.L. (Alessandra Lumini); software, L.N., A.F. and D.C.; writing—review and editing, A.L. (Alessandra Lumini), D.C., A.L. (Andrea Loreggia), A.F. and L.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Code and link for downloading the dataset are publicly available at https://github.com/LorisNanni, (accessed on 20 April 2022).

Acknowledgments: Through their GPU Grant Program, NVIDIA donated the TitanX GPU that was used to train the CNNs presented in this work.

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

References

1. Feng, D.; Haase-Schütz, C.; Rosenbaum, L.; Hertlein, H.; Glaeser, C.; Timm, F.; Wiesbeck, W.; Dietmayer, K. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 1341–1360. [CrossRef]
2. Brandao, P.; Zisimopoulos, O.; Mazomenos, E.; Ciuti, G.; Bernal, J.; Visentini-Scarzanella, M.; Menciassi, A.; Dario, P.; Koulaouzidis, A.; Arezzo, A.; et al. Towards a Computed-Aided Diagnosis System in Colonoscopy: Automatic Polyp Segmentation Using Convolution Neural Networks. *J. Med. Robot. Res.* **2018**, *3*, 1840002. [CrossRef]
3. Noh, H.; Hong, S.; Han, B. Learning deconvolution network for semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1520–1528. [CrossRef]
4. Li, L. Deep Residual Autoencoder with Multiscaling for Semantic Segmentation of Land-Use Images. *Remote Sens.* **2019**, *11*, 2142. [CrossRef]
5. Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *Computer Vision–ECCV, Proceeding of the 2018: 15th European Conference, Munich, Germany, 8–14 September 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 833–851. [CrossRef]
6. 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:2101.07172.
7. Dong, B.; Wang, W.; Li, J.; Fan, D.-P. Polyp-PVT: Polyp Segmentation with Pyramid Vision Transformers. *arXiv* **2021**, arXiv:2108.06932.
8. Jha, D.; Smedsrud, P.H.; Riegler, M.A.; Halvorsen, P.; de Lange, T.; Johansen, D.; Johansen, H.D. Kvasir-SEG: A Segmented Polyp Dataset. In *MultiMedia Modeling. MMM 2020*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2019; Volume 11962, pp. 451–462. [CrossRef]
9. Zhang, Y.; Liu, H.; Hu, Q. TransFuse: Fusing Transformers and CNNs for Medical Image Segmentation. *arXiv* **2021**, arXiv:2102.08005v2.
10. Kim, T.; Lee, H.; Kim, D. UACANet: Uncertainty Augmented Context Attention for Polyp Segmentation. In Proceedings of the 29th ACM International Conference on Multimedia, Chengdu, China, 20–24 October 2021; pp. 2167–2175. [CrossRef]
11. Minaee, S.; Boykov, Y.Y.; Porikli, F.; Plaza, A.J.; Kehtarnavaz, N.; Terzopoulos, D. Image Segmentation Using Deep Learning: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, 1. [CrossRef] [PubMed]
12. Lumini, A.; Nanni, L. Fair comparison of skin detection approaches on publicly available datasets. *Expert Syst. Appl.* **2020**, *160*, 113677. [CrossRef]
13. Roy, K.; Mohanty, A.; Sahay, R.R. Deep Learning Based Hand Detection in Cluttered Environment Using Skin Segmentation. In Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, Italy, 22–29 October 2017; pp. 640–649. [CrossRef]

14. Arsalan, M.; Kim, D.S.; Owais, M.; Park, K.R. OR-Skip-Net: Outer residual skip network for skin segmentation in non-ideal situations. *Expert Syst. Appl.* **2019**, *141*, 112922. [CrossRef]
15. Shahriar, S.; Siddiquee, A.; Islam, T.; Ghosh, A.; Chakraborty, R.; Khan, A.I.; Shahnaz, C.; Fattah, S.A. Real-Time American Sign Language Recognition Using Skin Segmentation and Image Category Classification with Convolutional Neural Network and Deep Learning. In Proceedings of the TENCON 2018—2018 IEEE Region 10 Conference, Jeju, Korea, 28–31 October 2018; pp. 1168–1171. [CrossRef]
16. Reena, M.R.; Ameer, P. Localization and recognition of leukocytes in peripheral blood: A deep learning approach. *Comput. Biol. Med.* **2020**, *126*, 104034. [CrossRef]
17. Liu, Y.; Cao, F.; Zhao, J.; Chu, J. Segmentation of White Blood Cells Image Using Adaptive Location and Iteration. *IEEE J. Biomed. Heal. Inform.* **2016**, *21*, 1644–1655. [CrossRef]
18. Shelhamer, E.; Long, J.; Darrell, T. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 640–651. [CrossRef]
19. 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]
20. 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.* **2018**, *40*, 834–848. [CrossRef] [PubMed]
21. 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. *arXiv* **2021**, arXiv:2012.15840.
22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
23. Khan, A.; Sohail, A.; Zahoora, U.; Qureshi, A.S. A survey of the recent architectures of deep convolutional neural networks. *Artif. Intell. Rev.* **2020**, *53*, 5455–5516. [CrossRef]
24. Nanni, L.; Cuza, D.; Lumini, A.; Loreggia, A.; Brahnam, S. Deep ensembles in bioimage segmentation. *arXiv* **2021**. [CrossRef]
25. Jadon, S. A survey of loss functions for semantic segmentation. In Proceedings of the 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), Via del Mar, Chile, 27–29 October 2020. [CrossRef]
26. Sudre, C.H.; Li, W.; Vercauteren, T.; Ourselin, S.; Cardoso, M.J. Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations. *arXiv* **2017**, arXiv:1707.03237.
27. Salehi, S.S.M.; Erdogmus, D.; Gholipour, A. Tversky loss function for image segmentation using 3D fully convolutional deep networks. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Cham, Switzerland, 2017; Volume 10541. [CrossRef]
28. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 318–327. [CrossRef] [PubMed]
29. Abraham, N.; Khan, N.M. A Novel Focal Tversky Loss Function with Improved Attention U-Net for Lesion Segmentation. In Proceedings of the 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), Venice, Italy, 8–11 April 2019. [CrossRef]
30. Qin, X.; Zhang, Z.; Huang, C.; Gao, C.; Dehghan, M.; Jagersand, M. BASNet: Boundary-Aware Salient Object Detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019. [CrossRef]
31. Wang, Z.; Bovik, A.; Sheikh, H.; Simoncelli, E. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]
32. Aurelio, Y.S.; de Almeida, G.M.; de Castro, C.L.; Braga, A.P. Learning from Imbalanced Data Sets with Weighted Cross-Entropy Function. *Neural Process. Lett.* **2019**, *50*, 1937–1949. [CrossRef]
33. Rahman, M.A.; Wang, Y. Optimizing intersection-over-union in deep neural networks for image segmentation. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Cham, Switzerland, 2016; pp. 234–244.
34. Cho, Y.-J. Weighted Intersection over Union (wIoU): A New Evaluation Metric for Image Segmentation. *arXiv* **2021**. [CrossRef]
35. Yang, D.; Roth, H.; Wang, X.; Xu, Z.; Myronenko, A.; Xu, D. Enhancing Foreground Boundaries for Medical Image Segmentation. *arXiv* **2020**. [CrossRef]
36. Chen, Z.; Zhou, H.; Lai, J.; Yang, L.; Xie, X. Contour-Aware Loss: Boundary-Aware Learning for Salient Object Segmentation. *IEEE Trans. Image Process.* **2020**, *30*, 431–443. [CrossRef]
37. Jha, D.; Ali, S.; Tomar, N.K.; Johansen, H.D.; Johansen, D.; Rittscher, J.; Riegler, M.A.; Halvorsen, P. Real-time polyp detection, localisation and segmentation in colonoscopy using deep learning. *IEEE Access.* **2021**, *9*, 40496–40510. [CrossRef] [PubMed]
38. Bernal, J.; Sánchez, J.; Vilariño, F. Towards automatic polyp detection with a polyp appearance model. *Pattern Recognit.* **2012**, *45*, 3166–3182. [CrossRef]
39. Vázquez, D.; Bernal, J.; Sánchez, F.J.; Fernández-Esparrach, M.G.; López, A.M.; Romero, A.; Drozdzal, M.; Courville, A. A Benchmark for Endoluminal Scene Segmentation of Colonoscopy Images. *J. Heal. Eng.* **2017**, *2017*, 1–9. [CrossRef] [PubMed]
40. Silva, J.S.; Histace, A.; Romain, O.; Dray, X.; Granado, B. Toward embedded detection of polyps in WCE images for early diagnosis of colorectal cancer. *Int. J. Comput. Assist. Radiol. Surg.* **2013**, *9*, 283–293. [CrossRef] [PubMed]

41. Bernal, J.; Sánchez, F.J.; Fernández-Esparrach, M.G.; Gil, D.; Rodríguez, C.; Vilariño, F. WM-DOVA maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians. *Comput. Med Imaging Graph.* **2015**, *43*, 99–111. [CrossRef]
42. Phung, S.; Bouzerdoum, A.; Chai, D. Skin segmentation using color pixel classification: Analysis and comparison. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 148–154. [CrossRef]
43. Tan, W.R.; Chan, C.S.; Yogarajah, P.; Condell, J. A Fusion Approach for Efficient Human Skin Detection. *IEEE Trans. Ind. Informatics* **2011**, *8*, 138–147. [CrossRef]
44. Huang, L.; Xia, T.; Zhang, Y.; Lin, S. Human skin detection in images by MSER analysis. In Proceedings of the 2011 18th IEEE International Conference on Image Processing, Brussels, Belgium, 11–14 September 2011; pp. 1257–1260. [CrossRef]
45. Ruiz-Del-Solar, J.; Verschae, R. Skin detection using neighborhood information. In Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Seoul, Korea, 19 May 2004; pp. 463–468. [CrossRef]
46. Jones, M.J.; Rehg, J. Statistical Color Models with Application to Skin Detection. *Int. J. Comput. Vis.* **2002**, *46*, 81–96. [CrossRef]
47. Casati, J.P.B.; Moraes, D.R.; Rodrigues, E.L.L. SFA: A human skin image database based on FERET and AR facial images. In Proceedings of the IX Workshop de Visão Computacional, Anais do VIII Workshop de Visão Computacional, Rio de Janeiro, Brazil, 3–5 June 2013.
48. Kawulok, M.; Kawulok, J.; Nalepa, J.; Smolka, B. Self-adaptive algorithm for segmenting skin regions. *EURASIP J. Adv. Signal Process.* **2014**, *2014*, 170. [CrossRef]
49. Schmugge, S.J.; Jayaram, S.; Shin, M.C.; Tsap, L.V. Objective evaluation of approaches of skin detection using ROC analysis. *Comput. Vis. Image Underst.* **2007**, *108*, 41–51. [CrossRef]
50. Miguel, J.C.S.; Suja, S. Skin detection by dual maximization of detectors agreement for video monitoring. *Pattern Recognit. Lett.* **2013**, *34*, 2102–2109. [CrossRef]
51. Abdallah, A.S.; El-Nasr, M.A.; Abbott, A.L. A new color image database for benchmarking of automatic face detection and human skin segmentation techniques. *World Acad. Sci. Eng. Technol.* **2007**, *36*, 38–42.
52. Wang, J.; Markert, K.; Everingham, M. Learning Models for Object Recognition from Natural Language Descriptions. *BMVC* **2009**. [CrossRef]
53. Zhao, P.; Li, C.; Rahaman, 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] [PubMed]
54. 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.
55. Wei, J.; Hu, Y.; Zhang, R.; Li, Z.; Zhou, S.K.; Cui, S. Shallow Attention Network for Polyp Segmentation. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Cham, Switzerland, 2021; pp. 699–708. [CrossRef]
56. Zhao, X.; Zhang, L.; Lu, H. Automatic Polyp Segmentation via Multi-scale Subtraction Network. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Cham, Switzerland, 2021; pp. 120–130. [CrossRef]
57. 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]
58. Song, P.; Li, J.; Fan, H. Attention based multi-scale parallel network for polyp segmentation. *Comput. Biol. Med.* **2022**, *146*, 105476. [CrossRef] [PubMed]
59. Filali, I.; Achour, B.; Belkadi, M.; Lalam, M. Graph ranking based butterfly segmentation in ecological images. *Ecol. Inform.* **2022**, *68*, 101553. [CrossRef]
60. Tang, H.; Wang, B.; Chen, X. Deep learning techniques for automatic butterfly segmentation in ecological images. *Comput. Electron. Agric.* **2020**, *178*, 105739. [CrossRef]
61. Pudil, P.; Novovičová, J.; Kittler, J. Floating search methods in feature selection. *Pattern Recognit. Lett.* **1994**, *15*, 1119–1125. [CrossRef]
62. Kuncheva, L.; Whitaker, C.J. Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy. *Mach. Learn.* **2003**, *51*, 181–207. [CrossRef]