doi:10.20944/preprints202107.0691.v1

Article

Deep ensembles based on Stochastic Activations for Semantic Segmentation

Alessandra Lumini[1,*], Loris Nanni[2] and Gianluca Maguolo[2]

[1] DISI, Università di Bologna, Via dell'università 50, 47521 Cesena, Italy; alessandra.lumini@unibo.it
[2] DEI, University of Padua, viale Gradenigo 6, Padua, Italy; loris.nanni@unipd.it
* Correspondence: alessandra.lumini@unibo.it

Abstract: Semantic segmentation is a very popular topic in modern computer vision and it has applications to many fields. Researchers proposed a variety of architectures over time, but the most common ones exploit an encoder-decoder structure that aims to capture the semantics of the image and it low level features. The encoder uses convolutional layers, in general with a stride larger than one, to extract the features, while the decoder recreates the image by upsampling an using skip connections with the first layers. In this work, we use DeepLab as architecture to test the effectiveness of creating an ensemble of networks by randomly changing the activation functions inside the network multiple times. We also use different backbone networks in our DeepLab to validate our findings. We manage to reach a dice coefficient of 0.888, and a mean Intersection over Union (mIoU) of 0.825, in the competitive Kvasir-SEG dataset. Results in skin detection also confirm the performance of the proposed ensemble, which is ranked first with respect to other state-of-the-art approaches (including HardNet) in a large set of testing datasets. The developed code will be available at https://github.com/LorisNanni

Keywords: semantic segmentation; activation function, deep ensembles.

1. Introduction

Semantic segmentation is a computer vision application that consists in labelling the pixels of an image with the class they belong to. This has very important applications in many fields such as autonomous driving [1] and computer aided medical diagnosis [2]. In recent years, deep learning techniques became the most relevant ones to address this problem. An early architecture for sematic segmentation was U-Net [3], which was based on an encoder-decoder structure. However, it failed to precisely classify the borders of the figures, due to the lack of skip-connections in the decoder. After that, many other segmentation networks were proposed and most of them followed more or less the same structure [4–6].

In this paper we use DeepLab v3+, which is one of the architectures of the DeepLab family. Here we focus on two applications, in particular on colorectal cancer segmentation and on skin detection.

Colorectal Cancer is one of the most dangerous cancers according to the statistics. The early diagnosis is crucial to be able to fully remove it while it is small. The presence of polyps in the colon is highly correlated with the appearance of cancer, hence they must be recognized and removed as soon as possible [7]. However, this is a challenging task even for trained doctors, hence an automatic tool able to recognize them would be very useful in this case. The boundaries of polyps are not always easy to recognize, due to their similarity with surrounding mucosa, besides there might be partial occlusions. Polyp belong to four different classes: adenoma, serrated, hyperplastic and mixed, which is quite rare. This makes the classification and the detection even harder.

Skin detection is a completely different task when it comes to the applications. It is a useful step for face detection, body tracking and gesture recognition [8–10]. However, the deep learning tools, as well as the challenges, are very similar in polyp segmentation and skin detection. Again, we usually face occlusions and intra-class variance, since images have very different light and the subjects can be very different from each other.

The outbreak of deep learning for computer vision led to an increasing effort to improve the classification performances of segmentators on a variety of applications. Nowadays segmentators have been reported to have performances comparable with human experts in polyp segmentation [2,11–13]. However, the tasks we are discussing are older and the first segmentators use classic machine learning techniques. For example, Thambawita et al. [14] trained five models for polyp segmentation including both classical machine learning techniques as well as convolutional networks. Guo et al. [15] proposed a couple of fully convolutional networks to participate at the Gastrointestinal Image ANAlysis (GIANA) in 2017 and 2018 and managed to reach the first and second place in the ranking in two consecutive years. Until recently, polyp segmentators were trained and tested on very small datasets, preventing the networks to generalize enough and also not allowing a good statistical significance of the results. Besides, many times the larger datasets were not publicly released [2,11].

Jha et al. [16] recently proposed a new public polyp dataset, which is called Kvasir-SEG dataset and it is made by 1,000 polyp images annotated at the Oslo University Hospital by expert endoscopists at pixel level. Jha et al. [17] managed to train a segmentator on this novel dataset based on ResNet and U-Net and managed to reach very promising results.

Skin segmentation also saw an increase in the number of papers dealing with this issue. For example, Phung et al. [18] proposed a segmentator based on classical machine learning techniques such as histograms analysis and Gaussian mixture classifiers. More recently, Roy et al. [19] proposed a system for hand recognition based on skin segmentation using CNNs. Arsalan et al. [20] used a CNN with skip connections for generic skin recognition, following the modern trends that tend to apply skip connections in neural networks more and more often. Shahriar et al. [21] used skin recognition again to detect hands, with the purpose of interpreting the sign language. They also used CNNs for their segmentation.

Our approach to segmentation consists in creating an ensemble of existing networks, but we randomly substitute their activation functions, which are usually ReLUs, with random ones extracted from a pool of activations proposed in the literature. We use different backbone networks for our DeepLab v3 architecture, which are ResNet, Xception, EfficentNet and MobileNet. Our aim is to show that we are able to train a large number of high performing classifiers that are independent enough from each other to be useful when included in an ensemble.

Each neural network has been stochastically designed by varying the activation layers in order to increase the diversity of the ensemble, then a pool of K most diverse networks has been selected (using only training data) to be included in the final ensemble. Experimental results carried out in two very different segmentation problems confirm the good performance of the ensemble. Our approach has been compared with other state-of-the-art methods in both problems, included the recently proposed HarDNet-MSEG [49] which here is evaluated for the first time in a skin segmentation problem.

2. Materials and Methods

2.1. *Deep Learning for Semantic Image Segmentation*

Image segmentation is a pixel-based classification problem which performs pixel-level labeling with a set of object categories for all image pixels. Fully Convolutional Networks (FNC) [6] are one of the first attempts to use CNN for segmentation: they were designed by replacing the last fully connected layers of a net with a fully-convolutional layer that allow the classification of the image on a per-pixel basis.

doi:10.20944/preprints202107.0691.v1

A step forward in the design of segmentation network is done by the encoder-decoder architecture [3] which overcomes the loss of information of FNC due to the absence of deconvolution, by proposing an architecture where a multi-layer deconvolution network is learned. A similar architecture is proposed by U-Net, a U-shape network where the decoder part downsamples the image and increases the number of features, while the opposite encoder part increases the image resolution to the input size [22]. Another encoder-decoder structure is proposed in SegNet [4], which uses VGG [23] as backbone encoder, coupled to a symmetric decoder structure. In SegNet decoding is performed using max pooling indices from the corresponding encoder layer, as opposed to concatenating it as in U-Net, thus saving memory and getting a better boundary reconstruction.

The next step to image segmentation is represented by DeepLab [24], a semantic segmentation model designed by Google which achieves dense prediction by simply up-sampling the output of the last convolution layer and computing pixel-wise loss. The novelty is in the use of atrous convolution for up-sample: it is a dilated convolution which uses a dilation rate to effectively enlarge the field of view of filters without increasing the number of parameters or the amount of computation. The last improvement of the DeepLab family is DeepLabV3+ [25], which combines cascaded and parallel modules of dilated convolutions and it is the architecture used in this work.

The DeepLab family of segmentators [24–27] is a very popular collection of segmentation tools. They have three key features. First, the used dilated convolutions to avoid the decrease in resolution caused by pooling layers and large strides. The second one is Atrous Spatial Pyramid Pooling, which consists in using filters with multiple sampling rates to get relevant information from the image at different scales. The third one is a better way to localize object boundaries that combines usual convolutional networks and probabilistic graphical models. DeepLabV3 was the third member of the DeepLab family and which consisted in an improvement of the previous versions by combining cascade and parallel modules of dilated convolutions. Besides, the Atrous Spatial Pyramid Pooling is modified adding a 1x1 convolution and batch normalization. The output of this network is given by a final layer which is again a 1x1 convolution that outputs the probability distribution of the classes on every pixel. The version that we use here, however, is DeepLabv3+, which is a modification of the older version that includes a decoder with point-wise convolutions, that operate on the same channel and on different locations, and depth-wise convolutions, that operate at the same location but on different channels. They kept the same encoder structure of DeepLabV3.

Several other architectures have been proposed in the literature for image segmentation, including recurrent neural network based models, attention-based models and generative models. The interested reader can refer to [28] for a recent survey.

Apart from the main architecture of the network, there are a handful of other good design choices that would help achieve good performance. For example, the choice of a pretrained backbone for the encoder part of the network. Among several CNNs [29] widely used for transfer learning we tested the following models:

- MobileNet-v2 [30] is a lightweight CNN designed for mobile devices based on depthwise separable convolutions.
- ResNet18 and ResNet50 [31] are two CNNs of the ResNet family, a set of architectures based on the use of residual blocks in which intermediate layers of a block learn a residual function with reference to the block input.
- Xception [32], is a CNN architecture that relies solely on depthwise separable convolution layers.
- IncR, Inception-ResNet-v2 [33] combines the Inception architecture with residual connections. In the Inception-Resnet block, multiple sized convolutional filters are combined with residual connections, replacing the filter concatenation stage of the Inception architecture.
- EfficentNetb0 [34], is a family of CNNs designed to scale well with performance. EfficientNet-B0 is a simple mobile-size baseline architecture, the other networks of the

family are obtained applying an effective compound scaling method for increasing the model size to achieve maximum accuracy gains.

In Table 1 a summary of the above models is reported.

Table 1: Summary of CNN models.

Network	Depth	Size (MB)	Parameters (Millions)	Input Size
mobilenetv2	53	13	3.5	224×224
resnet18	18	44	11.7	224×224
resnet50	50	96	25.6	224×224
xception	71	85	22.9	299×299
IncR	164	209	55.9	299×299
efficientnetb0	82	20	5.3	224×224

Also the choice of the loss function influences the way the network is trained. The most commonly used loss function for the task of image segmentation is a pixel-wise cross entropy loss. This loss treats the problem as a multi-class classification problem at pixel level comparing the class predictions to the actual label. Pixel-wise loss is calculated as the log loss summed over all the classes and averaged over all pixels. This can be a problem if some classes have unbalanced representation in the image, as training can be dominated by the most prevalent class. A possible solution is to use weighting for each class in order to counteract a class imbalance present in the dataset [6].

Another popular loss function for image segmentation is the Dice loss [35], which is based on the Sørensen-Dice similarity coefficient for measuring overlap between two segmented images. This measure ranges from 0 to 1 where a Dice coefficient of 1 denotes perfect and complete overlap. The dice loss is used in this work. Other popular loss functions for image segmentation and their usage for fast and better convergence of a model are reviewed in [36] .

Moreover, the choice of the activation function can be significant. ReLU is the nonlinearity that most works use in the area, but several works have reported improved results with different activation functions [37]. In subsection 2.2 our approach for perturbing models by replacing activation layer is explained.

Finally, data augmentation can help avoid overfitting since in many applications the size of the dataset is small compared to the number of parameters in a segmentation deep neural network. We perform experiments with data augmentation, consisting in horizontal and vertical flips and rotations of 90°.

2.2. *Stochastic Activation Selection*

Given a neural network architecture and a pool of different activation functions, Stochastic Activation Selection consists in creating different versions of the same architecture that differ in the choice of the activation layers. This method was first introduced in [37]. The process to create a new network is based on the replacement of each activation layer (ReLU) by a new activation function which can be fixed a priori or randomly selected from the ones in the pool. This new function is substituted into the original architecture. This leads to a new network, which in the stochastic version, has different activation layers through the network. Since this is a random procedure, it yields a different network every time. Hence, we iterate the process multiple times to create many different networks that we use to create an ensemble of neural networks. We train each network independently on the same set of data and then we merge their results using the sum rule, which consists in averaging the softmax output of all the networks in the ensemble.

In our paper, we use Deeplabv3+ [25] as neural architecture. The pool of activation functions is made by ReLU and a list of its modifications proposed in the literature: ReLU [38], Leaky ReLU [39], ELU [40], PReLU [41], S-Shaped ReLU [42] (SReLU), Adaptive Piecewise Linear Unit [43] (APLU), Mexican ReLU [44] (MeLU) (with $k \in \{4,8\}$), Gaussian

doi:10.20944/preprints202107.0691.v1

Linear Unit (GaLU) [37] (with $k \in \{4,8\}$), PDELU, [45], Swish (fixed and learnable) [46], Soft Root Sign [47], Mish (fixed and learnable) [48] and Soft Learnable [37].

3. Results on colorectal cancer segmentation

3.1 Datasets, testing protocol and metrics

All the experiments on colorectal cancer segmentation have been carried out on the Kvasir-SEG dataset [16] which includes 1000 polyp images acquired by a high-resolution electromagnetic imaging system, with a ground-truth consisting of bounding boxes and segmentation masks. For a fair comparison with other approaches (see table 4) as [17] and [49] we use the following testing protocol: 880 images are used for training, and the remaining 120 for testing.

The image sizes vary between 332 × 487 to 1920 × 1072 pixels. For training purposes, the images are resized to the input size of each model, but for performance evaluation the predicted masks are resized back to the original dimensions (please note that other approaches evaluated performance on the resized version of the images).

We train our models with SGD optimizer for 20 epochs and a learning rate of 10e-2 (see the code for details) using the Dice loss function and data augmentation.

Several metrics have been proposed in the literature to evaluate the performance of an image segmentation models. We report metrics for segmentation in two classes (foreground/background), which are suited to the polyp segmentation problem, anyway they can be easily extended to multiclass problems. The following metrics are the most popular to quantify model performance. All the following definitions hold for single images and are defined pixel-wise:

- Accuracy / Precision / Recall / F1-score / F2-score can be defined for a bi-class problem (or for each class in case of multiclass) starting from the confusion matrix (TP, TN, FP, FN refer to the true positives, true negatives, false positives and false negatives, respectively) as follows:

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}} \tag{1}$$

is the number of pixels correctly classified over the total number of pixels in the image.

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{2}$$

is the fraction of the polyp that is correctly classified.

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{3}$$

is the fraction of the model polyp outputs that were actually polyp pixels.

$$F1 - score = \frac{2 \cdot \mathrm{TP}}{2 \cdot \mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{4}$$

$$F2 - score = \frac{5 \cdot Precision \cdot Recall}{4 \cdot Precision + Recall} \tag{5}$$

are two measures that try to average precision and recall.

- Intersection over Union (IoU): IoU is defined as the area of intersection between the predicted segmentation map A and the ground truth map B, divided by the area of the union between the two maps:

$$IoU = \frac{|A \cap B|}{|A \cup B|} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{6}$$

- Dice: Dice coefficient is defined as twice the overlap area of the predicted and ground-truth maps divided by the total number of pixels. For binary maps, with foreground as the positive class, the Dice coefficient is identical to the F1-score:

$$Dice = \frac{|A \cap B|}{|A| + |B|} = \frac{2 \cdot \mathrm{TP}}{2 \cdot \mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{7}$$

All the above reported metrics range in [0,1] and must be maximized. The final performance is obtained averaging on the test set the performance obtained for each test image.

3.2 Experiments

The first experiment (**Table 2**) is aimed at comparing the different backbone networks listed in subsection 2.1. Since the size of images in Kvasir dataset is quite large we also evaluate versions of the Resnet with larger input size, i.e. 299×299 (resnet18-299/resnet50-299) and 352×352 (resnet18-352/resnet50-352).

Table 2: Experiments with different backbones.

Backbone	IoU	Dice	F2	Prec.	Rec.	Acc.
Mobilenetv2	0.734	0.823	0.827	0.863	0.841	0.947
resnet18	0.759	0.844	0.845	0.882	0.856	0.952
resnet50	0.751	0.837	0.836	0.883	0.845	0.952
xception	0.699	0.799	0.792	0.870	0.800	0.943
IncR	0.793	0.871	0.878	0.889	0.892	0.961
efficientnetb0	0.705	0.800	0.801	0.860	0.814	0.944
resnet18-299	0.782	0.863	0.870	0.881	0.883	0.959
resnet50-299	0.798	0.872	0.876	**0.898**	0.886	0.962
resnet18-352	0.787	0.865	0.871	0.891	0.884	0.960
resnet50-352	**0.801**	**0.872**	**0.884**	0.881	**0.900**	**0.964**

The second experiment (**Table 3**) is aimed at designing effective ensembles by varying the activation functions. Each ensemble is fusion by the sum rule of 14 models (since we use 14 activation functions). The ensemble name is the concatenation of the name of the backbone network and a string to identify the creation approach:

- act: each network is obtained by deterministically substituting each activation layer by one of the activation functions of subsection 2.2 (the same function for all the layers, but a different function for each network)
- sto: ensembles of stochastic models, whose activation layers have been replaced by a randomly selected activation function (which may be different for each layer)
- sel: ensembles of "selected" stochastic models. The network selection is performed using cross validation on the training set among 100 resnet50 stochastic models. The selection procedure is based on the idea of Sequential Forward Floating Selection (SFFS) [50], a selection method originally proposed for feature selection and here used for selecting the most performing/independent classifiers to be added to the ensemble. SFFS is an iterative method which, at each step, adds to the final ensemble the model which provides the highest incremental of performance to existing subset of models. Then a backtracking step is performed in order to exclude the worst model from the actual ensemble. Since SFFS requires a training phase we perform a 3-fold cross validation on the training set. For a fair comparison with other ensembles, we selected a set of 14 networks, which are finally fine-tuned on the whole augmented training set at larger resolution.

- relu: an ensemble of original models which differ only for the random initialization before training. It means that all the starting models in the ensemble are the same, except for the initialization.

Table 3: Experiments on ensembles.

Ensemble name	IoU	Dice	F2	Prec.	Rec.	Acc.
resnet18_act	0.774	0.856	0.856	0.888	0.867	0.955
resnet18_relu	0.774	0.858	0.858	0.892	0.867	0.955
resnet18_sto	0.780	0.860	0.857	0.898	0.864	0.956
resnet50_act	0.779	0.858	0.859	0.894	0.869	0.957
resnet50_relu	0.772	0.855	0.858	0.889	0.870	0.955
resnet50_sto	0.779	0.859	0.864	0.891	0.877	0.957
resnet50-352_sto	0.820	0.885	0.888	**0.915**	0.896	0.966
resnet50-352_sel	**0.825**	**0.888**	**0.892**	**0.915**	**0.902**	**0.967**

Finally, in Table 4 a comparison with some state-of-the-art results is reported.

Table 4: State-of-the-art approaches using the same testing protocol (all values are those reported in the reference paper, except for our approaches). The results of many methods are reported in [17], please read it for the original reference of a given approach. Other results [51][52] using a different protocol are not included in the comparison.

Method	IoU	Dice	F2	Prec.	Rec.	Acc.
resnet50-352	0.801	0.872	0.884	0.881	0.900	0.964
resnet50-352_sel	0.825	0.888	0.892	0.915	0.902	0.967
U-Net [17]	0.471	0.597	0.598	0.672	0.617	0.894
ResUNet [17]	0.572	0.69	0.699	0.745	0.725	0.917
ResUNet++ [17]	0.613	0.714	0.72	0.784	0.742	0.917

FCN8 [17]	0.737	0.831	0.825	0.882	0.835	0.952
HRNet [17]	0.759	0.845	0.847	0.878	0.859	0.952
DoubleUNet [17]	0.733	0.813	0.82	0.861	0.84	0.949
PSPNet [17]	0.744	0.841	0.831	0.89	0.836	0.953
DeepLabv3+ResNet50 [17]	0.776	0.857	0.855	0.891	0.861	0.961
DeepLabv3+ResNet101[17]	0.786	0.864	0.857	0.906	0.859	0.961
U-Net ResNet34 [17]	0.81	0.876	0.862	0.944	0.86	0.968
ColonSegNet [17]	0.724	0.821	0.821	0.843	0.850	0.949
DDANet [53]	0.78	0.858	---	0.864	0.888	---
HarDNet-MSEG [49]	0.848	0.904	0.915	0.907	0.923	0.969

4. Result on skin segmentation

4.1 Datasets, testing protocol and metrics

To evaluate the proposed ensemble for image segmentation we also perform a test on another relevant segmentation problem: skin segmentation. A skin segmentation (or detection) is a problem that discriminates regions in images and videos into the two classes skin and nonskin. Following the testing framework developed in reference [54], the performance results of the ensemble proposed here are compared to several state-of-the-art approaches on 11 datasets (Table 5) for skin segmentation; the training protocol provides that network models are trained only on the first 2000 images of the ECU dataset; while the other skin datasets are used only for testing (including the remaining 2000 from ECU).

Ta**ble 5.** Summary of the Skin segmentation datasets.

ShortName	Name	#Samples	Ref.
FV	Feeval Skin video DB	8991	[55]
Prat	Pratheepan	78	[56]
MCG	MCG-skin	1000	[57]
UC	UChile DB-skin	103	[58]
CMQ	Compaq	4675	[59]
SFA	SFA	1118	[60]
HGR	Hand Gesture Recognition	1558	[61]
Sch	Schmugge dataset	845	[62]
VMD	5 datasets for human activity recognition	285	[63]
ECU	ECU Face and Skin Detection	4000	[18]
VT	VT-AAST	66	[64]

The evaluation and comparison of the state-of-the-art approaches is performed according to the most used performance indicators in skin segmentation: F_1-measure, i.e. Dice, which is calculated at pixel-level (and not at image-level) to be independent on the image size in the different databases

4.2 Experiments

Table 6 reports the results of the evaluation of some of the networks and ensemble proposed in this paper compared to some state-of-the-art approaches on the testing sets described above. For each dataset the F1-measure is used as performance indicator, moreover the average F1-measure on the datasets is reported and the rank of the method with respect to the average value is calculated. The results of approaches followed by a citation are taken from the related papers, while for HarDNet are calculated using the same parameter configuration of the Polyp dataset [49] (a loss function which is a weighed sum of binary cross entropy and IoU, Adam optimizer with learning rate 0.001 and 100 epochs). As to our methods are concerned, in order to avoid overfitting we have maintained for the training on skin the same parameter configuration described above for polyp segmentation, including data augmentation, SGD optimizer with learning rate 0.1 for 20 epochs. Due to this configuration the results are quite different from those published in [54] for the same network, but the aim in this case was to validate our ensemble without an ad hoc tuning per dataset.

Table 6. Experiments on skin datasets (F1-measure). The last two columns report the average F1-measure on all the tested datasets and the rank of Avg.

Method	FV	Prat	MCG	UC	CMQ	SFA	HGR	Sch	VMD	ECU	VT	Avg	Rank
resnet50-224	0.694	0.874	0.862	0.866	0.797	0.939	0.954	0.760	0.608	0.927	0.682	0.815	7
resnet50-352	0.745	0.910	0.880	0.881	0.831	0.948	0.962	0.784	0.727	0.945	0.742	0.850	4
HarDNet-224	0.674	0.890	0.882	0.894	0.819	0.949	0.963	0.792	0.677	0.936	**0.756**	0.839	3
HarDNet-352	0.667	0.913	**0.887**	0.902	0.835	**0.952**	**0.968**	**0.795**	0.729	0.946	0.744	0.849	2
resnet50-352_sel	0.742	**0.917**	0.884	**0.910**	**0.840**	**0.952**	**0.968**	0.785	**0.742**	**0.949**	0.755	**0.859**	**1**
FusAct3 [37]	0.790	0.874	0.884	0.896	0.825	0.951	0.961	0.776	0.669	0.933	0.737	0.845	6
FusAct10 [37]	0.796	0.864	0.884	0.899	0.821	0.951	0.959	0.776	0.671	0.929	0.748	0.845	5
SegNet [54]	0.717	0.730	0.813	0.802	0.737	0.889	0.869	0.708	0.328	-	-	-	
U-Net [54]	0.576	0.787	0.779	0.713	0.686	0.848	0.836	0.671	0.332	-	-	-	
DeepLab [54]	**0.771**	0.875	0.879	0.899	0.817	0.939	0.954	0.774	0.628	-	-	-	

Compared with the state-of-the-art results in [54] and [37] (only the best method are reported in Table 6 for sake of space), the proposed ensemble resnet50-352_sel gets the best average performance. Notice that in [54] we have compared DeepLabV3+ with several other skin detector methods, it was shown that DeepLabV3+ obtained state of the art performance. This is a valuable result, since it proves that the good performance reported for the previous problem can be replicated in a very different context.

5. Discussion

Clearly using larger input sizes boost the performance of Resnet50 as proved by results in Table 2 for stand-alone models and Table 3 for ensembles. For ensemble creation, stochastic variation of activation functions (sto) allows a performance improvement with respect to a simple fusion of network based on ReLu activations (relu) or a set of networks differing by the activation function (act), moreover the selection procedure (sel) allows for a further improvement. In fact, the best performance, among the ensembles, is obtained by resnet50-352_sel (Table 3).

Our best approach obtains the best performance with except to HarDNet-MSEG [49] a segmentation network based on weighed loss. Notice that our approach strongly outperforms several other deep learning approaches including the recently published ColonSegNet [17] which works with larger image size (512).

We are aware that ensemble methods greatly increase computational costs and complexity with respect to a stand-alone network, therefore we suggest a simple rejection rule to reduce the computational effort. Considering that in a real dataset the incidence of images presenting polyps is quite low, we can use a first level rule to reject images non containing polyps based on a single net, and then use the ensemble only to gain a more precise segmentation if needed. Preliminary tests using a very low threshold, suggests that it is possible to set a rule able to discard images non containing lesions without losing in precision.

6. Conclusions

Semantic segmentation is a very important topic in medical-image analysis. In this paper our aim is to optimize the performance in polyp segmentation during colonoscopy examinations and skin detection.

We have compared several convolution neural network architectures, including ResNet, Xception, EfficentNet, MobileNet, HarDNet and different methods for building ensemble of CNN.

Our reported results show that the best ensemble obtains state of the art performance in the tested dataset (Kvasir-SEG dataset, skin test sets). To reproduce our results the MATLAB source code is available at GitHub: https://github.com/LorisNanni.

As a future work we plan to deal with the complexity problem of deep neural networks. Deploying large models or big ensembles on the edge is infeasible, since smartphones and IoT sensors are resource-constrained devices; hence, it is vital to focus research also on techniques for compressing large models into a compact one with minimal performance loss. We plan to study the feasibility of reduce the complexity of our ensemble by applying one of more of the following techniques: pruning, quantization, low-rank factorization and distillation.

Author Contributions: L.N. and A.L. conceived the presented idea., A.L. performed the experiments. G.M., A.L. and L.N. wrote the manuscript.

Acknowledgments: We gratefully acknowledge the support of NVIDIA Corporation for the “NVIDIA Hardware Donation Grant” of a Titan X used in this research.

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.
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.
3. Noh, H.; Hong, S.; Han, B. Learning deconvolution network for semantic segmentation. In Proceedings of the Proceedings of the IEEE International Conference on Computer Vision; 2015.
4. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, doi:10.1109/TPAMI.2016.2644615.
5. Bullock, J.; Cuesta-Lázaro, C.; Quera-Bofarull, A. XNet: A convolutional neural network (CNN) implementation for medical X-Ray image segmentation suitable for small datasets. In Proceedings of the Medical Imaging 2019: Biomedical Applications in Molecular, Structural, and Functional Imaging; 2019; Vol. 10953, p. 109531Z.
6. Shelhamer, E.; Long, J.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, doi:10.1109/TPAMI.2016.2572683.
7. Roncucci, L.; Mariani, F. Prevention of colorectal cancer: How many tools do we have in our basket? *Eur. J. Intern. Med.* 2015, *26*, 752–756.
8. Rein-Lien Hsu; Abdel-Mottaleb, M.; Jain, A.K. Face detection in color images. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 696–706, doi:10.1109/34.1000242.
9. Argyros, A.A.; Lourakis, M.I.A. Real-time tracking of multiple skin-colored objects with a possibly moving camera. *Lect.*

Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) **2004**, doi:10.1007/978-3-540-24672-5_29.

10. Han, J.; Award, G.M.; Sutherland, A.; Wu, H. Automatic skin segmentation for gesture recognition combining region and support vector machine active learning. In Proceedings of the Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition; 2006; pp. 237–242.
11. Wang, Y.; Tavanapong, W.; Wong, J.; Oh, J.; De Groen, P.C. Part-based multiderivative edge cross-sectional profiles for polyp detection in colonoscopy. *IEEE J. Biomed. Heal. Informatics* **2013**, *18*, 1379–1389.
12. Mori, Y.; Kudo, S.; Berzin, T.M.; Misawa, M.; Takeda, K. Computer-aided diagnosis for colonoscopy. *Endoscopy* **2017**, *49*, 813.
13. Wang, P.; Xiao, X.; Brown, J.R.G.; Berzin, T.M.; Tu, M.; Xiong, F.; Hu, X.; Liu, P.; Song, Y.; Zhang, D.; et al. Development and validation of a deep-learning algorithm for the detection of polyps during colonoscopy. *Nat. Biomed. Eng.* **2018**, *2*, 741–748.
14. Thambawita, V.; Jha, D.; Riegler, M.; Halvorsen, P.; Hammer, H.L.; Johansen, H.D.; Johansen, D. The medico-task 2018: Disease detection in the gastrointestinal tract using global features and deep learning. *arXiv Prepr. arXiv1810.13278* **2018**.
15. Guo, Y.B.; Matuszewski, B. Giana polyp segmentation with fully convolutional dilation neural networks. In Proceedings of the Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications; 2019; pp. 632–641.
16. 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 Proceedings of the Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); 2020.
17. Jha, D.; Ali, S.; Johansen, H.D.; Johansen, D.; Rittscher, J.; Riegler, M.A.; Halvorsen, P. Real-time polyp detection, localisation and segmentation in colonoscopy using deep learning. *arXiv* 2020.
18. Phung, S.L.; Bouzerdoum, A.; Chai, D. Skin segmentation using color pixel classification: Analysis and comparison. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 148–154, doi:10.1109/TPAMI.2005.17.
19. Roy, K.; Mohanty, A.; Sahay, R.R. Deep Learning Based Hand Detection in Cluttered Environment Using Skin Segmentation. In Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW); 2017; pp. 640–649.
20. 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.* **2020**, *141*, 112922, doi:10.1016/J.ESWA.2019.112922.
21. 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; 2018; pp. 1168–1171.
22. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); 2015.
23. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *Int. Conf. Learn. Represent.* **2015**, 1–14, doi:10.1016/j.infsof.2008.09.005.
24. 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**, doi:10.1109/TPAMI.2017.2699184.
25. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); 2018.
26. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv Prepr. arXiv1412.7062* **2014**.
27. Chen, L.-C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv Prepr. arXiv1706.05587* **2017**.
28. Minaee, S.; Boykov, Y.; Porikli, F.; Plaza, A.; Kehtarnavaz, N.; Terzopoulos, D. Image segmentation using deep learning: A

Survey. *arXiv* 2020.

29. Khan, A.; Sohail, A.; Zahoora, U.; Qureshi, A.S. A survey of the recent architectures of deep convolutional neural networks. *Artif. Intell. Rev.* **2020**, doi:10.1007/s10462-020-09825-6.
30. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2018.
31. 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); 2016; pp. 770–778.
32. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017; 2017.
33. Szegedy, C.; et al. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *Proc. IEEE Int. Conf. Comput. Vis.* **2017**, *115*, 4278–4284, doi:10.1145/3038912.3052569.
34. Tan, M.; Le, Q. V. EfficientNet: Rethinking model scaling for convolutional neural networks. In Proceedings of the 36th International Conference on Machine Learning, ICML 2019; 2019.
35. Sudre, C.H.; Li, W.; Vercauteren, T.; Ourselin, S.; Jorge Cardoso, M. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In Proceedings of the Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); 2017.
36. 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 2020; 2020.
37. Nanni, L.; Lumini, A.; Ghidoni, S.; Maguolo, G. Stochastic selection of activation layers for convolutional neural networks. *Sensors (Switzerland)* **2020**, doi:10.3390/s20061626.
38. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the Journal of Machine Learning Research; 2011.
39. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. In Proceedings of the in ICML Workshop on Deep Learning for Audio, Speech and Language Processing; 2013.
40. Clevert, D.A.; Unterthiner, T.; Hochreiter, S. Fast and accurate deep network learning by exponential linear units (ELUs). In Proceedings of the 4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings; 2016.
41. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the Proceedings of the IEEE International Conference on Computer Vision; 2015.
42. Jin, X.; Xu, C.; Feng, J.; Wei, Y.; Xiong, J.; Yan, S. Deep learning with S-shaped rectified linear activation units. In Proceedings of the 30th AAAI Conference on Artificial Intelligence, AAAI 2016; 2016.
43. Agostinelli, F.; Hoffman, M.; Sadowski, P.; Baldi, P. Learning activation functions to improve deep neural networks. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015 - Workshop Track Proceedings; 2015.
44. Maguolo, G.; Nanni, L.; Ghidoni, S. Ensemble of Convolutional Neural Networks Trained with Different Activation Functions. *CoRR* **2019**, *abs/1905.0*.
45. Cheng, Q.; Li, H.; Wu, Q.; Ma, L.; King, N.N. Parametric Deformable Exponential Linear Units for deep neural networks. *Neural Networks* **2020**.
46. Ramachandran, P.; Zoph, B.; Le, Q. V Searching for Activation Functions. *CoRR* **2017**, *abs/1710.05941*.
47. Zhou, Y.; Li, D.; Huo, S.; Kung, S.-Y. Soft-Root-Sign Activation Function. **2020**.
48. Misra, D. Mish: A self regularized non-monotonic neural activation function. *arXiv Prepr. arXiv1908.08681* **2019**.
49. 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**.
50. Pudil, P.; Novovičová, J.; Kittler, J. Floating search methods in feature selection. *Pattern Recognit. Lett.* **1994**, doi:10.1016/0167-

8655(94)90127-9.

51. Safarov, S.; Whangbo, T.K. A-denseunet: Adaptive densely connected unet for polyp segmentation in colonoscopy images with atrous convolution. *Sensors* **2021**, doi:10.3390/s21041441.

52. Branch, M.V.L.; Carvalho, A.S. Polyp Segmentation in Colonoscopy Images using U-Net-MobileNetV2. **2021**.

53. Tomar, N.K.; Jha, D.; Ali, S.; Johansen, H.D.; Johansen, D.; Riegler, M.A.; Halvorsen, P. DDANet: Dual Decoder Attention Network for Automatic Polyp Segmentation. In; 2021 ISBN 9783030687922.

54. Lumini, A.; Nanni, L. Fair comparison of skin detection approaches on publicly available datasets. *Expert Syst. Appl.* 2020.

55. Stöttinger, J.; Hanbury, A.; Liensberger, C.; Khan, R. Skin paths for contextual flagging adult videos. In Proceedings of the Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); 2009; Vol. 5876 LNCS, pp. 303–314.

56. Tan, W.R.; Chan, C.S.; Yogarajah, P.; Condell, J. A Fusion Approach for Efficient Human Skin Detection. *Ind. Informatics, IEEE Trans.* **2012**, *8*, 138–147, doi:10.1109/TII.2011.2172451.

57. Huang, L.; Xia, T.; Zhang, Y.; Lin, S. Human skin detection in images by MSER analysis. *18th IEEE Int. Conf. Image Process.* **2011**, 1257–1260, doi:10.1109/ICIP.2011.6115661.

58. Ruiz-Del-Solar, J.; Verschae, R. Skin detection using neighborhood information. In Proceedings of the Proceedings - Sixth IEEE International Conference on Automatic Face and Gesture Recognition; 2004; pp. 463–468.

59. Jones, M.J.; Rehg, J.M. Statistical color models with application to skin detection. *Int. J. Comput. Vis.* **2002**, *46*, 81–96, doi:10.1023/A:1013200319198.

60. 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 Visao Computational, Rio de Janeiro; 2013.

61. Kawulok, M.; Kawulok, J.; Nalepa, J.; Smolka, B. Self-adaptive algorithm for segmenting skin regions. *EURASIP J. Adv. Signal Process.* **2014**, 1–22, doi:10.1186/1687-6180-2014-170.

62. 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, doi:10.1016/j.cviu.2006.10.009.

63. Sanmiguel, J.C.; Suja, S. Skin detection by dual maximization of detectors agreement for video monitoring. *Pattern Recognit. Lett.* **2013**, *34*, 2102–2109, doi:10.1016/j.patrec.2013.07.016.

64. 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. In Proceedings of the Proceedings of World Academy of Science, Engineering and Technology; 2007; Vol. 20, pp. 353–357.