Peer-reviewed version available at *Sensors* **2020**, *20*, 1626; doi:10.3390/s20061626

Article

Stochastic Activation Function Layers for Convolutional Neural Networks

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

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

Abstract: In recent years, the field of deep learning achieved considerable success in pattern recognition, image segmentation and may other classification fields. There are a lot of studies and practical applications of deep learning on images, video or text classification. In this study, we suggest a method for changing the architecture of the most performing CNN models with the aim of designing new models to be used as stand-alone networks or as a component of an ensemble. We propose to replace each activation layer of a CNN (usually a ReLu layer) by a different activation function stochastically drawn from a set of activation functions: in this way the resulting CNN has a different set of activation function layers.

The code developed for this work will be available at https://github.com/LorisNanni

Keywords: Convolutional Neural Networks; ensemble of classifiers; activation functions; image classification; skin detection.

1. Introduction

Deep neural networks have become extremely popular as they achieve state-of-the-art performance on a variety of important applications including image classification, image segmentation, language processing and computer vision [1]. Deep neural networks typically have a set of linear components whose parameters are usually learned to fit the data, and a set of nonlinearities, which are pre-specified, typically in the form of a sigmoid, a tanh function, a rectified linear unit, or a max-pooling function. The presence of non-linear activation functions at each neuron is essential to give the network the ability of approximate arbitrarily complex functions [2], and its choice affects both the speed of training and net accuracy. The design of new activation functions in order that improve training speed and network accuracy is an active area of research [3][4]. Recently, the sigmoid and hyperbolic tangent, which were the most widely used activations functions, have been replaced by Rectified Linear Units (ReLU) to train deep networks [5]: ReLU is a piecewise linear function equivalent to the identity for positive inputs and zero for negative ones. Thanks to the good performance of ReLU and the fact that it is fast, effective, and simple to evaluate, several alternatives to the standard ReLu function have been proposed in the literature. The most known "fixed" activation function are: Leaky ReLU [6] an activation function equal to ReLU for positive inputs but having a very small slope $\alpha > 0$ for negative ones, ELU [4] which exponentially decreases to a limit point α in the negative space, SELU [7], a scaled version of ELU (by a constant λ). Together with "fixed" activation functions, several "learnable" activation functions have been proposed as: Parametric ReLU (PReLU) [8] which is a Leaky ReLU where the amount of α is learned during training, Adaptive Piecewise Linear Unit (APLU) [3] which is a piecewise linear activation with learnable parameters, Swish, one of the best performing functions according to [9], which is the combination of a sigmoid function and a trainable parameter, and the recent Mexican ReLU (MeLU) [10] which is a piecewise linear activation function that is the sum of PReLU and multiple Mexican hat functions.

In this paper, we perform a largescale empirical comparison of different activation function across a variety of image classification task and for an image segmentation problem. Starting form two of the best performing models, i.e. ResNet50 [11] for the classification task and DeepLab-v3 [12] for the segmentation task, we compare different approaches for replacing ReLu layers and different methods for building ensembles of CNNs obtained by varying the activation function layers.

After comparing several activation functions, we propose to design a new model based on the combination of different activation functions at different levels of the graph: to this aim, we propose a method for stochastic selection of activation functions to replace each ReLu layer of the starting network. The activation functions are randomly selected from a set of 9 approaches, including the most effective ones. After training the new models on the target problem, they are fused together to build an ensemble of CNNs.

The proposed framework for ensemble creation is evaluated on two different applications: image classification and image segmentation. In the image classification field, we deal with several medical problems including in our benchmark 13 image classification datasets. CNNs have already been used on several medical datasets reaching very high performance, including keratinocyte carcinomas and malignant melanomas detection [13], thyroid nodules classification [14] from ultrasound images or breast cancer recognition [15]. Our testing protocol include a fine-tuning of each model in each dataset and a testing evaluation and comparison: our experiments show that the proposed method works well in all the tested problems gaining state-of-the-art classification performance [16].

In the image segmentation field we deal with the skin segmentation problem: the discrimination of skin and non-skin regions in a digital image has a wide range of applications including face detection [17], body tracking [18], gesture recognition [19], objectionable content filtering [20]. Skin detection has great relevance also in the medical field, where it is employed as a component of face detection or body tracking: for example in the remote photoplethysmography (rPPG) problem [21] it is a component of a system based at solving the problem of estimating the heart rate of a subject given a video stream of his/her face. In our experiment, we carry out a comparison of several approaches performing a single training on a small dataset including only 2000 labeled images, while testing is performed on 11 different datasets including images from very different applications. The reported results show that the proposed method is able to reach state of the art performance [22] in most of the benchmark datasets even without ad-hoc tuning.

2. Materials and Methods

In this section we describe both the starting models and the stochastic method proposes to design a new CNN models. CNNs are deep neural networks designed to work similarly to the human brain in visually perceiving the world. They are made of several type of "layers" on neurons: i.e. convolutional layers, activation layers, subsampling layers, fully connected layers [23]. In particular, activation layers are aimed at deciding if a neuron would fire or not according to a nonlinear transformation of the input signal. Since activation functions play a vital role in the training of CNN, several activation functions have been proposed: in this work we evaluate some variant of the standard ReLU function, which will be discussed in the next section.

In the literature several CNN architecture have been proposed both for the image classification (i.e. AlexNet [24], GoogleNet [25], InceptionV3 [26], VGGNet [27], ResNet [11], DenseNet [28]) and segmentation problems (i.e. SegNet [29], U-Net [30], Deeplabv3+ [12]). In our experiments, we select two of the most performing models: i.e. ResNet50 [11] for image classification and Deeplabv3+ [12] for segmentation. ResNet50 is a 50-layer network, which introduces a new "network-in-network" architecture using residual layers. ResNet50, which was the winner of ILSVRC 2015, is one of the most performing and popular architecture used for image classification. In our experiments, all the models for image classification have been fine-tuned on the training set of each classification problem according to the following options: batch size 32, learning rate 0.0001, max epoch 30 (data augmentation based random reflection on both axes and two independent random rescales of both axes by two factors uniformly sampled in [1,2] has been used in all the epochs).

Peer-reviewed version available at Sensors 2020, 20, 1626; doi:10.3390/s20061626

For image segmentation purposes we select Deeplabv3+ [12], a recent architecture based on atrous convolution, in which the filter is not applied to all adjacent pixels of an image but rather a spaced-out lattice of pixels. Deeplabv3+ uses four parallel atrous convolutions (each with differing atrous rates) followed by a "Pyramid Pooling" method. Since DeepLabv3+ is based on encoder-decoder structure, it can be built on top of a powerful pretrained CNN architecture: in this work, we selected again ResNet50 for this task, anyway our internal evaluation showed that ResNet101 and ResNet 34 gained similar performance. All the models for skin segmentation have been trained on a small dataset of 2000 images using the same options: batch size 32, learning rate 0.001, max epoch 50 (data augmentation has been used only in the first 30 epochs), class weighting.

This study considers 10 different activation functions (more details, and specific reference for each function, are given in [10]), namely the widely used ReLU and several variants. The functions used are summarized in the following, together with their derivatives.
The well-known ReLU activation function is defined as:

$$y_i = f(x_i) = \begin{cases} 0, & x_i < 0 \\ x_i, & x_i \geq 0, \end{cases}$$

and its derivative is easily evaluated as:

$$\frac{dy_i}{dx_i} = f'(x_i) = \begin{cases} 0, & x_i < 0 \\ 1, & x_i \geq 0. \end{cases}$$

This work also considers several variants of the original ReLU function. The first variant is the Leaky ReLU function, defined as:

$$y_i = f(x_i) = \begin{cases} ax_i, & x_i < 0 \\ x_i, & x_i \geq 0, \end{cases}$$

where a is a small real number (0.01 in this study). The main advantage of Leaky ReLU is that the gradient is always positive (no point has a zero gradient):

$$\frac{dy_i}{dx_i} = f'(x_i) = \begin{cases} a, & x_i < 0 \\ 1, & x_i \geq 0. \end{cases}$$

The second variant of the ReLU function considered in this work is the Exponential Linear Unit (ELU), which is defined as:

$$y_i = f(x_i) = \begin{cases} a(\exp(x_i) - 1), & x_i < 0 \\ x_i, & x_i \geq 0, \end{cases}$$

where a is a real number (1 in this study). ELU has a gradient that is always positive:

$$\frac{dy_i}{dx_i} = f'(x_i) = \begin{cases} a \exp(x_i), & x_i < 0 \\ 1, & x_i \geq 0. \end{cases}$$

The third variant of ReLU is the Scaled Exponential Linear Unit (SELU). This is defined as:

$$y_i = f(x_i) = \begin{cases} sa(\exp x_i - 1), & x_i < 0 \\ sx_i, & x_i \geq 0, \end{cases}$$

where a and s are real numbers – in our case $a = 1.6733$ and $s = 1.0507$. SELU is very similar to ELU with the additional scaling parameter s, which can be used to correct the gradient when it is exploding or vanishing. The gradient in this case is given by:

$$\frac{dy_i}{dx_i} = f'(x_i) = \begin{cases} sa \exp(x_i), & x_i < 0 \\ s, & x_i \geq 0. \end{cases}$$

The Parametric ReLU (PReLU) is the fourth variant that is considered here. It is defined by:

$$y_i = f(x_i) = \begin{cases} a_c x_i, & x_i < 0 \\ x_i, & x_i \geq 0, \end{cases}$$

where ac is a set of real numbers, one for each input channel. PReLU is similar to Leaky ReLU, the only difference being that the a_c parameters are learned. The gradient of PReLU is:

$$\frac{dy_i}{dx_i} = f'(x_i) = \begin{cases} a_c, & x_i < 0 \\ 1, & x_i \geq 0 \end{cases} \text{ and } \frac{dy_i}{da_c} = \begin{cases} x_i, & x_i < 0 \\ 0, & x_i \geq 0. \end{cases}$$

S-Shaped ReLU (SReLU) is the fifth variant. It is defined as a piecewise linear function:

$$y_i = f(x_i) = \begin{cases} t^l + a^l(x_i - t^l), & x_i < t^l \\ x_i, & t^l \le x_i \le t^r \\ t^r + a^r(x_i - t^r), & x_i > t^r \end{cases}.$$

In this case four learnable parameters are used, t^l, t^r, a^l, and a^r, expressed as real numbers. They are initialized to $a^l = 0, t^l = 0$, $t^r = maxInput$, where $maxInput$ is a hyperparameter. SReLU is highly flexible thanks to the rather large number of tunable parameters. The gradients are given by:

$$\frac{dy_i}{dx_i} = f'(x_i) = \begin{cases} a^l, & x_i < t^l \\ 1, & t^l \le x_i \le t^r \\ a^r, & x_i > t^r \end{cases},$$

$$\frac{dy_i}{da^l} = \begin{cases} x_i - t^l, & x_i < t^l \\ 0, & x_i \ge t^l \end{cases}, \text{ and}$$

$$\frac{dy_i}{dt^l} = \begin{cases} -a^l, & x_i < t^l \\ 0, & x_i \ge t^l \end{cases}.$$

The sixth variant is APLU, namely the Adaptive Piecewise Linear Unit. As the name suggests, it is a linear piecewise function. It is defined as:

$$y_i = \text{ReLU}(x_i) + \sum_{c=1}^{n} a_c \min(0, -x_i + b_c),$$

where a_c and b_c are real numbers, one for each input channel. The gradients are evaluated as:

$$\frac{df(x,a)}{da_c} = \begin{cases} -x + b_c, & x < b_c \\ 0, & x \ge b_c \end{cases} \text{ and } \frac{df(x,a)}{db_c} = \begin{cases} -a_c, & x < b_c \\ 0, & x \ge b_c \end{cases}.$$

In our tests the parameters a_c are initialized to 0, and the points are randomly chosen. We also added an L^2-penalty of 0.001 to the norm of the parameters a_c.

An interesting variant is the MeLU, that is, the Mexican ReLU, derived from the Mexican hat functions. These are defined as:

$$\phi^{a,\lambda}(x) = \max(\lambda - |x - a|, 0),$$

where a and λ are real numbers. These functions are used to define the MeLU function:

$$y_i = MeLU(x_i) = PReLU^{c_0}(x_i) + \sum_{j=1}^{k-1} c_j \, \phi^{a_j,\lambda_j}(x_i).$$

The parameter k represents the number of learnable parameters for each input channel, c_j are the learnable parameters, c_0 is the parameter vector in PReLU, and a_j and λ_j are fixed parameters chosen recursively. The MeLU activation function has interesting properties, inherited from the Mexican hat functions, that are continuous and piecewise differentiable. ReLU can be seen as a special case of MeLU, when all the c_i parameters are set to 0. This is important because pre-trained networks based on the ReLU function can be enhanced in a simple way using MeLU. Similar substitutions can be made when the source network is based on Leaky ReLU and PReLU.

As previously observed, MeLU is based on a set of learnable parameters. The number of parameters is sensibly higher with respect to SReLU and APLU, making MeLU more adaptable and with a higher representation power but more likely to overfit. The gradient is given by the Mexican hat functions. The MeLU activation function also has a positive impact on the optimization stage.

In our work the learnable parameters are initialized to 0, meaning that the MeLU starts as a plain ReLU function; the peculiar properties of the MeLU function come into play at a later stage of the training.

The Gaussian ReLU, also called GaLU, is the last activation function considered in our work. Its definition is based on the gaussian type functions:

$$\phi_g{}^{a,\lambda}(x) = \max(\lambda - |x - a|, 0) + \min(|x - a - 2\lambda| - \lambda, 0),$$

where a and λ are real numbers. The GaLU activation function is defined as:

$$y_i = GaLU(x_i) = PReLU^{c_0}(x_i) + \sum_{j=1}^{k-1} c_j \, \phi_g{}^{a_j,\lambda_j}(x_i),$$

which is a formulation similar to the one provided for MeLU, which again depends on the parameters a_j and λ_j. Again, the function is defined in this way to provide a good approximation of nonlinear functions.

3. Results

In order to evaluate the different activation functions detailed in section 2 and to validate the stochastic method for ensemble creation, we performed experiments on 13 well-known medical datasets for image classification and 11 datasets for skin segmentation. Table 1 summarizes the 13 datasets including a short abbreviation, the dataset name, the number of samples and classes and the testing protocol. We used in 12 out 13 datasets five-fold cross-validation (5CV), while we maintain a three fold division for the Laryngeal dataset (same protocol of [31]). Table 2 summarizes the 11 datasets used for skin segmentation. All the models have been trained only on the first 2000 images of the ECU dataset [32], therefore all the other skin dataset are used only for testing (for ECU only the last 2000 images not included in the training set have been used for testing).

The evaluation and comparison of the proposed approaches is performed according to two of the most used performance indicators in image classification and skin segmentation: accuracy and F_1-measure, respectively. Accuracy is the ratio between the number of true predictions and the total number of samples, while the F_1-measure is the harmonic mean of precision and recall and it is calculated according to the following formula $F_1 = 2tp/(2tp + fn + fp)$, where *tn*, *fn*,*tp*, *fp* are the number of true negatives, false negatives, true positives and false positives evaluated at pixel-level. According to other works in skin detection, F_1 is calculated at pixel-level instead of at image-level to be independent on the image size in the different databases. Finally to validate the experiments the Wilcoxon signed rank test [33] has been used.

Table 1. Summary of the Medical Datasets for image classification: Short Name (ShortN), Name, number of classes (#C), number of samples (#S), testing protocol, URL for downloading, Reference.

ShortN	Name	#C	#S	Protocol	URL for Download	Ref
CH	Chinese hamster ovary cells	5	327	5CV	http://ome.grc.nia.nih.gov/iicbu2008/hela/index.html#cho	[34]
HE	2D HELA	10	862	5CV	http://ome.grc.nia.nih.gov/iicbu2008/hela/index.html	[34]
LO	Locate Endogenous	10	502	5CV	http://locate.imb.uq.edu.au/downloads.shtml	[35]
TR	Locate Transfected	11	553	5CV	http://locate.imb.uq.edu.au/downloads.shtml	[35]
RN	Fly Cell	10	200	5CV	http://ome.grc.nia.nih.gov/iicbu2008/rnai/index.html	[36]
TB	Terminal bulb aging	7	970	5CV	https://ome.grc.nia.nih.gov/iicbu2008	[36]
LY	Lymphoma	3	375	5CV	https://ome.grc.nia.nih.gov/iicbu2008	[36]
MA	Muscle aging	4	237	5CV	https://ome.grc.nia.nih.gov/iicbu2008	[36]
LG	Liver gender	2	265	5CV	https://ome.grc.nia.nih.gov/iicbu2008	[36]
LA	Liver aging	4	529	5CV	https://ome.grc.nia.nih.gov/iicbu2008	[36]
CO	Human colorectal cancer	8	5000	5CV	https://zenodo.org/record/53169#.WaXjW8hJaUm	[37]
BGR	Breast grading carcinoma	3	300	5CV	https://zenodo.org/record/834910#.Wp1bQ-jOWUl	[38]
LAR	Laryngeal dataset	3	1320	Tr-Te	https://zenodo.org/record/1003200#.WdeQcnBx0nQ	[31]

Table 2. Summary of the Skin detection datasets with ground truth used for image segmentation: Short Name (ShortN), Name, number of images (#S), URL for downloading, Reference.

ShortN	Name	#S	URL for Download	Ref
CMQ	Compaq	4675	ask to the authors	[39]
UC	UChile DB-skin	103	http://agami.die.uchile.cl/skindiff/ (currently not available)	[40]
ECU	ECU Face and Skin Detection	4000	http://www.uow.edu.au/~phung/download.html (currently not available)	[32]
Sch	Schmugge dataset	845	https://www.researchgate.net/publication/257620282_skin_image_Data_set_with_ground_truth	[41]
FV	Feeval Skin video DB	8991	http://www.feeval.org/Data-sets/Skin_Colors.html	[42]
MCG	MCG-skin	1000	http://mcg.ict.ac.cn/result_data_02mcg_skin.html (ask to the authors)	[43]
Prat	Pratheepan	78	http://web.fsktm.um.edu.my/~cschan/downloads_skin_dataset.html	[44]
VMD	5 datasets for human activity recognition	285	http://www-vpu.eps.uam.es/publications/SkinDetDM/	[45]
SFA	SFA	1118	http://www1.sel.eesc.usp.br/sfa/	[46]
HGR	Hand Gesture Recognition	1558	http://sun.aei.polsl.pl/~mkawulok/gestures/	[47]
VT	VT-AAST	66	http://abdoaast.wixsite.com/abdallahabdallah/the-vt-mena-benchmarking-datas	[48]

In the first experiment, we evaluate the proposed methods for image classification on the datasets listed above: in **Table 3** the performance, in terms of accuracy, of several variant of ResNet50 obtained by varying the activation function and some the following ensembles are reported:

- The method named *ReLu* is our baseline, since it corresponds to the standard implementation of ResNet50. *ReLu* performs very well, but it is not the best performing activation function: many activation functions with INPUT parameter 255 work better than *ReLu* on average.
- It is a very valuable result the methods as *wMelu*(255), *MeLu*(255) and some other stand-alone approaches outperform *ReLu* in a large selection of classification problems. Starting from a model pretrained with ReLu and changing its activation layers, we obtained a sensible error reduction.
- It is difficult to select a function that win in all problems, therefore a good method to improve performance is to create an ensemble of different models: both *FusAct10* and *FusAct10*(255) work better than each of their single components.
- Designing the model by means of stochastic activation functions (i.e. *Random* or *Random*(255)) gives valuable results above all in the creation of ensembles: indeed both *FusRan10* and *FusRan10*(255) performs very well when compared to all stand-alone models and also to the ensembles above. *FusRan10*(255) is the first ranked method tested in these experiments.
- The two lightweight ensembles *FusAct3* and *FusRan3* strongly outperform stand-alone approaches and gain performance comparable with other heavier ensembles (composed by 10 or 20 models).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 February 2020 doi:10.20944/preprints202002.0231.v1

Peer-reviewed version available at Sensors 2020, 20, 1626; doi:10.3390/s20061626

Table 3. Performance of the proposed approaches in the medical image datasets (accuracy).

Method	Dataset CH	HE	LO	TR	RN	TB	LY	MA	LG	LA	CO	BG	LAR	Avg	Rank
ReLu	93.5	89.9	95.6	90.0	55.0	58.5	77.9	90.0	93.0	85.1	94.9	88.7	87.1	84.55	14
leakyReLu	89.2	87.1	92.8	84.2	34.0	57.1	70.9	79.2	93.7	82.5	95.7	90.3	87.3	80.30	21
SeLu	90.2	86.7	94.0	85.8	48.0	60.8	65.3	85.0	96.0	90.1	95.1	89.3	89.9	82.80	20
SReLu	91.4	85.6	92.6	83.3	30.0	55.9	69.3	75.0	88.0	82.1	95.7	89.0	89.5	79.02	23
APLu	92.3	87.1	93.2	80.9	25.0	54.1	67.2	76.7	93.0	82.7	95.5	90.3	88.9	78.99	24
GaLu	92.9	88.4	92.2	90.4	41.5	57.8	73.6	89.2	92.7	88.8	94.9	90.3	90.0	83.28	19
sGaLu	92.3	87.9	93.2	91.1	52.0	60.0	72.5	90.0	95.3	87.4	95.4	87.7	88.8	84.13	16
preLu	92.0	85.4	91.4	81.6	33.5	57.1	68.8	76.3	88.3	82.1	95.7	88.7	89.6	79.26	22
MeLu	91.1	85.4	92.8	84.9	27.5	55.4	68.5	77.1	90.0	79.4	95.3	89.3	87.2	78.76	26
wMeLu	92.9	86.4	91.8	82.9	25.5	56.3	67.5	76.3	91.0	82.5	94.8	89.7	88.8	78.95	25
SReLu(255)	92.3	89.4	93.0	90.7	56.5	59.7	73.3	91.7	98.3	89.0	95.5	89.7	87.9	85.15	12
APLu(255)	95.1	89.2	93.6	90.7	47.5	56.9	75.2	89.2	97.3	87.1	95.7	89.7	89.5	84.35	15
GaLu(255)	92.9	87.2	92.0	91.3	47.5	60.1	74.1	87.9	96.0	86.9	95.6	89.3	87.7	83.73	18
sGaLu(255)	93.5	87.8	95.6	89.8	55.0	63.1	76.0	90.4	95.0	85.3	95.1	89.7	89.8	85.09	13
MeLu(255)	92.9	90.2	95.0	91.8	57.0	59.8	78.4	87.5	97.3	85.1	95.7	89.3	88.3	85.26	10
wMeLu(255)	94.5	89.3	94.2	92.2	54.0	61.9	75.7	89.2	97.0	88.6	95.6	87.7	88.7	85.27	9
Random	90.2	90.0	94.2	91.6	54.5	62.0	77.3	90.8	95.7	90.5	95.1	89.0	87.1	85.23	11
Random(255)	93.2	88.5	94.4	91.6	51.5	59.1	73.9	88.3	94.0	89.1	95.1	86.7	88.0	84.11	17
FusAct10	93.5	90.7	**97.2**	92.7	56.0	63.9	77.6	90.8	96.3	91.4	96.4	90.0	90.0	86.67	7
FusAct10(255)	95.1	**91.3**	96.2	94.2	**63.0**	64.9	78.7	92.5	97.7	87.6	96.5	89.7	89.8	87.46	5
FusRan10	95.4	**91.3**	95.8	95.1	**63.0**	64.2	78.9	**93.8**	**98.7**	91.1	96.5	90.3	90.2	88.02	4
FusRan10(255)	**96.9**	91.2	96.8	**96.2**	58.5	**66.6**	**79.7**	92.5	98.3	**91.6**	**96.6**	89.7	**91.1**	**88.13**	1
Fus20	95.7	90.8	97.0	94.4	61.5	64.1	79.5	**93.8**	98.3	91.4	96.6	**91.0**	90.5	88.04	3
Fus20(255)	96.3	91.2	96.6	95.3	62.0	64.9	79.5	**93.8**	98.3	90.1	**96.6**	90.3	90.8	88.12	2
FusAct3(255)	93.9	91.5	94.8	93.1	58.5	63.5	77.6	91.3	98.3	88.0	96.3	89.0	89.4	86.55	8
FusRan3(255)	96.3	90.9	95.6	95.1	54.0	62.9	78.7	92.5	**98.7**	90.9	96.2	90.0	90.5	87.10	6

From the results in **Table 3** we can draw the following conclusions:

- The method named *ReLu* is our baseline, since it corresponds to the standard implementation of ResNet50. *ReLu* performs very well, but it is not the best performing activation function: many activation functions with INPUT parameter 255 work better than *ReLu* on average.
- It is a very valuable result the methods as *wMelu*(255), *MeLu*(255) and some other stand-alone approaches strongly outperform *ReLu* in a large selection of classification problems. Starting from a model pretrained with ReLu and changing its activation layers, we obtained a sensible error reduction. This mean that our approaches permit to boost the performance of the original ResNet50 in a large set of problems.
- It is difficult to select a function that win in all problems, therefore a good method to improve performance is to create an ensemble of different models: both *FusAct* and *FusAct*(255) work better than each of their single components.
- Designing the model by means of stochastic activation functions (i.e. *Random* or *Random*(255)) gives valuable results above all in the creation of ensembles: indeed both *FusRan10* and *FusRan10*(255) performs very well compared to all stand-alone models and also the ensembles above. *FusRan10*(255) is the first ranked method tested in these experiments.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 February 2020 doi:10.20944/preprints202002.0231.v1

Peer-reviewed version available at Sensors 2020, 20, 1626; doi:10.3390/s20061626

- The two small ensemble *FusAct3*(255) and *FusRan3*(255) performs very well strongly outperform stand-alone approaches and gain performance comparable with other heavier ensembles (composed by 10 or 20 models).

In the second experiment, we evaluate the proposed methods for skin segmentation on the 11 datasets listed above: in **Table 4** the performance of several variant of ResNet50 and some the following method/ensembles are reported in terms of F_1-measure:

- *ReLu* is the standard DeepLabv3+ segmentation CNN based on ResNet50 encoder. This net has shown state-of-the-art performance for skin segmentation [22].
- *leakyReLu, SeLu, SReLu, APLu, GaLu, sGaLu, preLu, MeLu, wMeLu* are the 9 activation functions used in the ResNet50 architecture. Some of them depends on a parameter MAXINPUT which has been set to 1 if not explicitly indicated in parenthesis (255).
- *FusAct10* and *FusAct10*(255) are the ensembles obtained by the fusion of all the 10 models listed above (fixing MAXINPUT=1 or 255)
- *FusAct3* is a lightweight ensemble obtained by the fusion of the best 3 stand-alone models (evaluated on the training set), i.e. *FusAct3= wMeLu + MeLu + preLu*
- *Random* and *Random*(*255*) are two stand-alone models designed according to stochastic selection method of activation functions described in section 2.
- *FusRan10* and *FusRan10*(255) are two ensembles obtained by the fusion of 10 stochastic models as *Random* or *Random*(255)
- *FusRan3* is the ensemble obtained by the fusion of 3 stochastic models as *Random*
- *Fus20 = FusAct10 + FusRan10*
- *Fus20*(255) = *FusAct10*(255) + *FusRan10*(255)

For each dataset, the best result is highlighted and in the last two columns report the average F_1-measure and the rank (calculated on the average F_1).

Table 4. Performance of the proposed approaches in the skin datasets (F1-measure).

	Dataset											Avg	Rank
Method	**FV**	**Prat**	**MCG**	**UC**	**CMQ**	**SFA**	**HGR**	**Sch**	**VMD**	**ECU**	**VT**		
ReLu	0.759	0.831	0.872	0.881	0.799	0.946	0.950	0.763	0.592	0.917	0.745	0.823	17
leakyReLu	0.753	0.853	0.876	0.875	0.804	0.944	0.955	0.762	0.606	0.921	0.716	0.824	13
SeLu	0.682	0.838	0.870	0.834	0.791	0.941	0.944	0.763	0.540	0.918	0.677	0.800	**26**
SReLu	0.722	0.839	0.867	0.860	0.807	0.950	0.958	0.743	0.610	0.919	0.709	0.817	24
APLu	0.774	0.840	0.874	0.880	0.796	0.942	0.945	0.761	0.593	0.914	0.745	0.824	15
GaLu	0.759	0.827	0.867	0.872	0.795	0.941	0.933	0.755	0.562	0.913	0.731	0.814	25
sGaLu	0.779	0.834	0.872	0.867	0.798	0.946	0.951	0.766	0.597	0.915	0.739	0.824	14
preLu	0.785	0.852	0.878	0.886	0.809	0.947	0.953	0.770	0.633	0.924	0.740	0.834	9
MeLu	0.768	0.861	0.878	0.879	0.819	0.947	0.953	0.768	0.643	0.927	0.725	0.834	10
wMeLu	0.768	0.869	0.878	0.888	0.821	0.945	0.956	0.771	0.616	0.929	0.706	0.832	11
SReLu(255)	0.758	0.831	0.872	0.879	0.797	0.946	0.949	0.764	0.592	0.916	0.744	0.823	18
APLu(255)	0.755	0.839	0.873	0.873	0.797	0.940	0.947	0.760	0.584	0.909	0.744	0.820	21
GaLu(255)	0.776	0.832	0.870	0.869	0.790	0.938	0.940	0.758	0.566	0.911	0.756	0.819	23
sGaLu(255)	0.769	0.845	0.876	0.886	0.797	0.944	0.951	0.764	0.617	0.919	0.741	0.828	12
MeLu(255)	0.757	0.836	0.874	0.872	0.792	0.943	0.944	0.767	0.570	0.913	0.744	0.819	22
wMeLu(255)	0.759	0.832	0.873	0.880	0.799	0.946	0.950	0.763	0.599	0.917	0.742	0.824	16
Random	0.757	0.852	0.876	0.889	0.804	0.937	0.947	0.764	0.569	0.920	0.730	0.822	19
Random(255)	0.732	0.844	0.873	0.878	0.797	0.944	0.937	0.758	0.595	0.914	0.751	0.820	20
FusAct10	0.796	0.864	0.884	0.899	0.821	**0.951**	0.959	0.776	0.671	0.929	0.748	0.845	2
FusAct10(255)	0.791	0.854	0.881	0.897	0.813	0.949	0.955	0.774	0.654	0.925	**0.761**	0.841	7
FusRan10	0.795	0.864	0.883	0.901	0.818	0.949	0.958	0.775	0.667	0.927	0.752	0.844	6
FusRan10(255)	**0.800**	0.867	0.884	**0.906**	0.819	0.950	0.958	**0.779**	0.655	0.927	0.749	0.845	4
Fus20	0.799	0.865	0.884	0.901	0.820	0.951	0.959	0.776	**0.673**	0.929	0.751	**0.846**	1

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 February 2020 doi:10.20944/preprints202002.0231.v1

Peer-reviewed version available at Sensors 2020, 20, 1626; doi:10.3390/s20061626

Fus20(255)	0.798	0.862	0.883	0.903	0.817	0.950	0.957	0.777	0.660	0.927	0.758	0.845	5
FusAct3	0.790	**0.874**	**0.884**	0.896	**0.825**	0.951	**0.961**	0.776	0.669	**0.933**	0.737	0.845	3
FusRan3	0.783	0.870	0.883	0.902	0.818	0.951	0.959	0.778	0.635	0.930	0.717	0.839	8

From the results in **Table 4** it is clear that:

- In this problem the activation functions with INPUT parameter 1 work better than those initialized by 255, therefore we fixed to 1 the INPUT for the ensembles with 3 models (FusAct3 and FusRan3.
- Similarly, to the image classification experiment, the ensembles work better than stand-alone approaches: *Fus20* is the best ranked method in our experiments, but two "lighter" ensembles as *FusAct3* and *FusAct10* gain very high performance.
- As in the classification problem, our approaches outperform *ReLu*, i.e. the standard DeepLabv3+ based on ResNet50, a state of the art approach for image segmentation.
- The reported results show that the proposed ensemble methods are able to reach state of the art performance [22] in most of the benchmark datasets.

Finally, we report some comparisons considering the Wilcoxon signed rank test, see Table 5 and Table 6. Notice that we report different approaches in Table 5 and Table 6, in each table we report performance of the most interesting approach for classification and segmentation (one approach for each size of ensembles). The reported p-values confirm the conclusions drawn from Tables 3 and 4.

Table 5. P-value of the comparison among some tested approaches in the medical image classification experiment (< denotes that the method in row wins, ^ denotes that the method in column wins, = denotes that there are were no statistically significant differences).

Classification	**ReLu**	**wMeLu(255)**	**FusRan3(255)**	**FusRan10(255)**	**Fus20(255)**
ReLu	---	^0.0046	^0.0210	^0.002	^0.002
wMeLu(255)		---	^0.0024	^0.004	^0.002
FusRan3(255)			---	^0.004	^0.004
FusRan10(255)				---	=0.5684
Fus20(255)					---

Table 6. P-value of the comparison among some tested approaches in the skin segmentation experiment.

Skin segmentation	**ReLu**	**preLu**	**FusAct3**	**FusAct10**	**Fus20**
ReLu	---	^0.0059	^0.0029	^0.001	^0.001
preLu		---	^0.0020	^0.001	^0.001
FusAct3			---	=0.9844	=0.6797
FusAct10				---	^0.0938
Fus20					---

Finally, notice that using a GTX1080 the classification time of ResNet is 0.024 seconds / image; this mean that also using an ensemble of 20 CNNs it is possible to classify two images each second using a single GTX1080.

4. Conclusions

In this study, we proposed a method for CNN model design based on changing the architecture of the most performing CNN models by stochastic layer replacement. We proposed to replace each activation layer of a CNN by a different activation function stochastically drawn from a given set: in such a way that the resulting model has different activation function layers. This generation process introduces diversity among models making them suitable for ensemble creation. Interestingly, this design approach has gained very strong performance for ensemble creation: a set of ResNet50-like models designed using stochastic replacement of ReLu layers and combined by sum rule strongly outperforms both standard ResNet50 and a single stochastic ResNet50 in our experiments. A large experimental evaluation, carried out in a wide set of benchmark problems both for image classification and image segmentation, showed that our idea can be used for building a high performance ensemble of CNNs.

Even if these first results are limited to a single, although performing model, we plan as a future work to evaluate the proposed method on a large class of models including lighter architectures suitable for mobile devices. The difficulty of studies involving ensembles of CNNs lies in the enormous speed and memory resources required to conduct such experiments.

Another research direction is related to the selection of the initial set of activation function according to their performance.

Author Contributions: Conceptualization, A.L. and L.N.; methodology, L.N.; software, A.L., S.G. and L.N.; validation, G.M..; formal analysis, A.L.; writing—original draft preparation, A.L. and L.N.; writing—review and editing, S.G. and G.M.. All authors have read and agreed to the published version of 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. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press, 2016;

2. Cho, Y.; Saul, L.K. Large-margin classification in infinite neural networks. *Neural Comput.* 2010.

3. 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.

4. 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.

5. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the Journal of Machine Learning Research; 2011.

6. 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.

7. Klambauer, G.; Unterthiner, T.; Mayr, A.; Hochreiter, S. Self-Normalizing Neural Networks. In Proceedings of the NIPS; 2017.

8. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on

imagenet classification. *Proc. IEEE Int. Conf. Comput. Vis.* **2015**, *2015 Inter*, 1026–1034.

9. Ramachandran, P.; Barret, Z.; Le, Q. V. Searching for activation functions. In Proceedings of the 6th International Conference on Learning Representations, ICLR 2018 - Workshop Track Proceedings; 2018.

10. Maguolo, G.; Nanni, L.; Ghidoni, S. Ensemble of Convolutional Neural Networks Trained with Different Activation Functions. **2019**.

11. 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.

12. 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.

13. Esteva, A.; Kuprel, B.; Novoa, R.A.; Ko, J.; Swetter, S.M.; Blau, H.M.; Thrun, S. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **2017**, *542*, 115–118.

14. Chi, J.; Walia, E.; Babyn, P.; Wang, J.; Groot, G.; Eramian, M. Thyroid nodule classification in ultrasound images by fine-tuning deep convolutional neural network. *J. Digit. Imaging* **2017**, *30*, 477–486.

15. Byra, M. Discriminant analysis of neural style representations for breast lesion classification in ultrasound. *Biocybern. Biomed. Eng.* **2018**, *38*, 684–690.

16. Nanni, L.; Brahnam, S.; Ghidoni, S.; Lumini, A. Bioimage Classification with Handcrafted and Learned Features. *IEEE/ACM Trans. Comput. Biol. Bioinforma.* **2018**.

17. Hsu, R.L.; Abdel-Mottaleb, M.; Jain, A.K. Face detection in color images. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**.

18. 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**.

19. 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.

20. Lee, J.-S.; Kuo, Y.-M.; Chung, P.-C.; Chen, E.-L. Naked image detection based on adaptive and extensible skin color model. *Pattern Recognit.* **2007**, *40*, 2261–2270.

21. Paracchini, M.; Marcon, M.; Villa, F.; Tubaro, S. Deep Skin Detection on Low Resolution Grayscale Images. *Pattern Recognit. Lett.* **2020**.

22. Lumini, A.; Nanni, L. Fair comparison of skin detection approaches on publicly available datasets. *arXiv Prepr. arXiv1802.02531* **2018**.

23. Liu, W.; Wang, Z.; Liu, X.; Zeng, N.; Liu, Y.; Alsaadi, F.E. A survey of deep neural network architectures and their applications. *Neurocomputing* **2017**.

24. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* **2012**, 1–9.

25. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2015; Vol. 07-12-June, pp. 1–9.

26. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016; Vol. 00, pp. 2818–2826.

27. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *Int. Conf. Learn. Represent.* **2015**, 1–14.

28. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017; 2017.

29. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**.

30. 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.

31. Moccia, S.; De Momi, E.; Guarnaschelli, M.; Savazzi, M.; Laborai, A.; Guastini, L.; Peretti, G.; Mattos, L.S. Confident texture-based laryngeal tissue classification for early stage diagnosis support. *J. Med. Imaging* **2017**, *4*, 34502.

32. 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.

33. Demšar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.

34. Boland, M. V; Murphy, R.F. A neural network classifier capable of recognizing the patterns of all major subcellular structures in fluorescence microscope images of HeLa cells. *Bioinformatics* **2001**, *17*, 1213–1223.

35. Hamilton, N.A.; Pantelic, R.S.; Hanson, K.; Teasdale, R.D. Fast automated cell phenotype image classification. *BMC Bioinformatics* **2007**, *8*, 110.

36. Shamir, L.; Orlov, N.; Mark Eckley, D.; Macura, T.J.; Goldberg, I.G. IICBU 2008: A proposed benchmark suite for biological image analysis. *Med. Biol. Eng. Comput.* **2008**, *46*, 943–947.

37. Kather, J.N.; Weis, C.-A.; Bianconi, F.; Melchers, S.M.; Schad, L.R.; Gaiser, T.; Marx, A.; Zöllner, F.G. Multi-class texture analysis in colorectal cancer histology. *Sci. Rep.* **2016**, *6*, 27988.

38. Dimitropoulos, K.; Barmpoutis, P.; Zioga, C.; Kamas, A.; Patsiaoura, K.; Grammalidis, N. Grading of invasive breast carcinoma through Grassmannian VLAD encoding. *PLoS One* **2017**, *12*, 1–18.

39. Jones, M.J.; Rehg, J.M. Statistical color models with application to skin detection. *Int. J. Comput. Vis.* **2002**, *46*, 81–96.

40. 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.

41. 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.

42. 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.

43. 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.

44. 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.

45. Sanmiguel, J.C.; Suja, S. Skin detection by dual maximization of detectors agreement for video monitoring. *Pattern Recognit. Lett.* **2013**, *34*, 2102–2109.

46. 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.

47. Kawulok, M.; Kawulok, J.; Nalepa, J.; Smolka, B. Self-adaptive algorithm for segmenting skin regions. *EURASIP J. Adv. Signal Process.* **2014**, 1–22.

48. 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.