Article

Not peer-reviewed version

Comparison of Different Methods for Building Ensembles of Convolutional Neural Networks

Loris Nanni [*] , Andrea Loreggia , Sheryl Brahnam

Posted Date: 4 September 2023

doi: 10.20944/preprints202309.0058.v1

Keywords: convolutional neural networks; ensembles; fusion

Article

Comparison of Different Methods for Building Ensembles of Convolutional Neural Networks

Loris Nanni [1], Andrea Loreggia [2] and Sheryl Brahnam [3]

[1] Department of Information Engineering, University of Padova, Padova, Italy; loris.nanni@unipd.it
[2] Department of Information Engineering, University of Brescia, Brescia, Italy; andrea.loreggia@unibs.it
[3] Information Technology and Cybersecurity, Missouri State University, 901 S. National, Springfield MO, 65804, USA; SBrahnam@missouristate.edu
* Correspondence: loris.nanni@unipd.it

Abstract: In computer vision and image analysis, Convolutional Neural Networks (CNNs) and other deep learning models are at the forefront of research and development. These advanced models have proven to be highly effective in tasks related to computer vision. One technique that has gained prominence in recent years is the construction of ensembles using Deep CNNs. These ensembles typically involve combining multiple pre-trained CNNs to create a more powerful and robust network. The purpose of this study is to evaluate the effectiveness of building CNN ensembles by combining several advanced techniques. Tested here are CNN ensembles constructed by replacing ReLU layers with different activation functions, employing various data augmentation techniques, and utilizing several algorithms, including some novel ones, that perturb network weights. Experimental results performed across many data sets representing different tasks demonstrate that our proposed methods for building deep ensembles produces superior results. All the resources required to replicate our experiments are available at https://github.com/LorisNanni.

Keywords: convolutional neural networks; ensembles; fusion

1. Introduction

Artificial neural networks (ANNs), which were initially developed in the 1950s, have had a checkered history, at times appreciated for their unique computational capabilities and at other times disparaged for being no better than statistical methods. Opinions shifted about a decade ago with deep neural networks, whose performance swiftly overshadowed that of other learners across various scientific, medical, and engineering domains. The prowess of deep learners is especially exemplified by the remarkable achievements of Convolutional Neural Networks (CNNs), one of the most renowned and robust deep learning architectures.

CNNs have consistently outperformed other classifiers in numerous applications, particularly in image recognition competitions where they frequently emerge as winners [1]. Not only do CNNs surpass traditional classifiers, but they also often outperform the recognition abilities of human beings. In the medical field, for instance, CNNs have been shown to perform better than human experts in detecting skin cancer [2,3], skin lesions on the face and scalp, and esophageal cancer (e.g., [4]). These successes, unsurprisingly, have led to a surge in imaging research investigating CNNs and other deep learners. For example, in the medical field, deep learners have become state-of-the-art at diagnosing diabetic retinopathy [5], Alzheimer's disease [6], skin detection [7], gastrointestinal ulcers, and a host of different cancers, as evidenced in recent reviews and studies.

CNNs, however, have limitations. It is widely recognized that they require many samples to avoid overfitting [8]. Acquiring image collections numbering in the hundreds of thousands for proper CNN training is an enormous enterprise [9]. In certain medical domains, it is prohibitively labor-intensive and costly [10]. Several well-established techniques have been developed to address the issue of overfitting with limited data, the two most common being transfer learning using pre-trained CNNs and data augmentation [11,12]. The literature is abundant with studies investigating both methods, and it has been observed that combining the two yields better results (e.g., [13]).

In addition to transfer learning and data augmentation, another powerful technique for enhancing the performance of deep learners generally, as well as on small sample sizes, is to construct ensembles of pre-trained CNNs [14]. Many robust CNN ensembles have been reported in recent years, showcasing their effectiveness in various applications. For instance, in [15], a deep CNN ensemble was developed for classifying ER (Estrogen Receptor) status from DCE-MRI (Dynamic Contrast-Enhanced Magnetic Resonance Imaging) breast volumes. In [16], the authors focused on diabetic muscular edema diagnosis and employed a hierarchical ensemble approach. In [17], a CNN ensemble was designed for whole-brain segmentation, while in [18] an ensemble approach for small lesion detection was proposed. In each of these cases, the ensemble was shown to perform better than standalone CNNs.

The success of ensemble learning involves combining the outputs of multiple classifiers [19] that introduce some degree of diversity, which can be achieved through various means [20]. One common approach is to train each CNN on different subsets of the available data [21]. Other approaches involve combining different types of CNN architectures [22], varying network depth [23,24], and introducing different activation functions [19]. Related work using these different approaches is addressed in Section 2.

This research paper focuses on image classification using ensembles of CNNs built with one network, ResNet50 [25], chosen because of its balanced tradeoff between performance and training time. The goal of this work is to perform an exhaustive evaluation across many data sets of the performance of combining three ensembling methods: 1) replacing ReLU layers with twenty different activation functions, 2) applying various data augmentation techniques, and 3) utilizing several methods to perturb network weights. Results of our experiments demonstrate that combining ensembles built with these methods produces superior results.

The contributions of this study include:

- An in-depth comparison and evaluation of three methods for building CNN ensembles, both standalone and in combination, verified across five different data sets;
- The introduction of several new methods for perturbing network weights;
- Free access to all resources, including the MATLAB source code, used in our experiments.

The remainder of this paper is organized as follows. Section 2 provides a review of the literature on building ensembles for CNNs. Full details about our ensemble are provided in Section 3. Section 4 presents and discusses the results of our experiments. Section 5 provides some final remarks and outlines research opportunities for the future.

2. Related Work

The related work in this field explores various strategies for creating ensembles of CNNs, with a focus on achieving high performance and maximizing the independence of predictions. Already addressed in the introduction are approaches based on training networks with different architectures and activation functions and using diverse training sets and data augmentation approaches for the same network architecture. In addition, ensembles can be generated by combining multiple pre-trained CNNs, employing various training algorithms, and applying distinct rules for combining networks.

The most intuitive approach to form a diverse ensemble involves training different models on the entire dataset. It can be challenging, however, to find models with similar performance, and employing a large number of models may only yield marginal improvements over the best model's performance. Conversely, the predictions of high-performing models might exhibit strong correlations, leading to an ensemble that performs on par with a single model.

Most researchers taking this intuitive approach primarily fine-tune or train well-known architectures from scratch, average the results, and then demonstrate through experiments that the ensemble outperforms individual stand-alone networks. For instance, in [26], Kassani et al. employed an ensemble of VGG19 [27], MobileNet [28], and DenseNet [29] to classify histopathological biopsies, showing that the ensemble consistently achieved better performance than each individual network across four different datasets. Similarly, Qummar et al. [5] proposed an ensemble comprising ResNet50 [30], Inception v3 [31], Xception [32], DenseNet121, and DenseNet169 [29] to detect diabetic retinopathy.

In their study, Liu et al. [33] constructed an ensemble comprising three distinct CNNs proposed in their paper and averaged their results. Their ensemble achieved higher accuracy than the best individual model on the FER2013 dataset [34]. Similarly, Kumar *et al.* [35] introduced an ensemble of pretrained AlexNet and GoogleNet [36] models from ImageNet, which were then fine-tuned on the ImageCLEF 2016 collection dataset [37]. They utilized the features extracted from the last fully connected layers of these networks to train an SVM, an approach that outperformed CNN baselines and remained competitive with state-of-the-art methods at that time. Pandey *et al.* [38] proposed FoodNet, an ensemble composed of finetuned AlexNet, GoogleNet, and ResNet50 models designed for food image recognition. The output features from these models were concatenated and passed through a fully connected layer and softmax classifier.

Utilizing diverse training sets to train a classifier proves to be an effective approach in generating semi-independent classifiers. This can be achieved through various methods, with one classic technique being bagging [39–41]. Bagging involves creating m training sets of size n from a larger training set by randomly selecting samples with uniform probability and with replacement. Subsequently, the same model is trained on each of these training sets. Examples of this approach to building ensembles include the work of Kim et al. [42], who proposed a bagging-based approach to train three distinct CNNs for vehicle type classification. Similarly, Dong et al. [43] applied bagging and CNNs to improve short-term load forecasting in smart grid, resulting in a significant reduction of the mean absolute percentage error (MAPE) from 33.47 to 28.51. As another example, Guo et al. [44] employed eight different datasets to train eight distinct networks for object detection. These datasets were formed by combining existing datasets in various ways. Remarkably, this straightforward approach led to substantial performance gains compared to individual models and brought the ensembles's performance close to the state-of-the-art on competitive datasets like COCO 2012.

The training algorithms employed in CNNs follow stochastic trajectories and operate on stochastic data batches. Consequently, training the same network multiple times may lead to different outcomes at the end of the process. To enhance the diversity among final models, they can be trained using distinct training algorithms. For instance, the authors in [45] constructed an ensemble for facial expression recognition using soft-label perturbation, where different losses were propagated for different samples. Similarly, Antiov et al. [46] utilized different network initializations to train multiple networks for gender predictions from face images.

Another approach to building ensembles is to adopt the same architecture but vary the activation functions. This can be done in a set of CNNs or within different layers of a single CNN [47]. One way to implement the latter approach is to select a random activation function from a pool for each layer in the original network [48]. The diversity introduced in this way makes activation functions an excellent candidate for generating ensembles of deep learners, as noted in [49], and is the tactic adopted in this work.

Finally, ensembles can vary in the selection of rules for merging results. A straightforward approach is majority voting, where the predominant output selected by the majority of the networks is taken [50–53]. Another common technique frequently cited in the literature is to average the softmax outputs of the networks [54–56]. More intricate methods have been proposed, such as that proposed in Lumini et al. [57], which calculates a learned weighted average of the softmax output.

3. Methods

The ensembles in this study are constructed using sets of ResNet50 architectures pre-trained on the ImageNet dataset, famous for its large scale. Since winning the ILSVRC 2015 contest, ResNet50 has gained in popularity and is well understood. It is particularly known for its skip connections, which allow the input of a block to be added to its output. This technique promotes gradient propagation and facilitates the flow of lower-level information to higher-level layers.

The training process in this work involves training each network with a batch size (BS) of 30 and a learning rate (LR) of 0.001 for 20 epochs. It is worth noting that the last fully connected layer has a learning rate 20 times larger than the rest of the layers. Ensemble decisions are combined using the average rule. This means that the softmax probabilities generated by each network in the ensemble for a given sample are averaged, resulting in a new score that is used for classification.

In the remainder of this section, we describe the methods we use to create ensembles of ResNet50.

3.1. Activation Functions

In this study, a set of more than twenty activation functions is investigated in building CNN ensembles. These activation functions include such well-known ones as ReLU, Leaky ReLU, ELU, SELU, PReLU, APLU, SReLU, MeLU, Splash, Mish, PDELU, Swish, Soft Learnable, etc. (the complete details/list of these activation functions can be found in [47]).

The main advantage of complex activation functions with learnable parameters is their ability to capture abstract features through nonlinear transformations, a characteristic commonly observed in shallow networks [58]. However, a potential drawback lies in their complexity: multiple learnable parameters require large data sets for training.

The stochastic approach mentioned in [47] is used to alter the activation functions in ResNet50. This method involves randomly replacing all activations within a network with a new activation function selected from a pool of potential candidates. The random selection process is repeated multiple times to generate a set of networks that are fused together in the ensemble. The performance of a pool of candidate activation functions varies depending on the specific CNN architecture. What this means is that some activation functions will perform poorly with ResNet50, while others will perform well. The result is significant variance among the ensemble members.

In the experimental section, the stochastic method of combining CNNs is referred to as "SE." It is important to note that the proposed ensemble approach does not pose a risk of overfitting. The replacement of activation functions is performed randomly without any ad hoc selection of specific datasets. Overfitting could potentially occur if the activation functions were chosen based on ad hoc data sets, but this is not the case in the proposed ensemble method.

3.2. Data Augmentation

During the training process, sets of networks are trained using different data augmentation techniques [11]. The following data augmentation methods have been utilized:

- APP1: This augmentation generates three new images based on a given image. It randomly reflects the image vertically and horizontally, resulting in two new images. The third transformation involves linearly scaling the original image along both axes with two factors randomly selected from a uniform distribution ranging from 1 to 2.
- APP2: Building upon APP1, this augmentation generates six new images. It includes the transformations of APP1 and adds three additional manipulations. First, image rotation is applied with a random angle extracted from the range of -10 to 10 degrees. Second, translation is performed by shifting the image along both axes with values randomly sampled from the interval of 0 to 5 pixels. Last, shear transformation is applied, with vertical and horizontal angles randomly selected from the range of 0 to 30 degrees.
- APP3: This augmentation replicates APP2 but excludes the shear and the scale transformations, resulting in four new images.
- APP4: This augmentation approach generates three new images by applying a transform based on Principal Component Analysis (PCA). The PCA coefficients extracted from a given image are subjected to three perturbations that generate three new images. For the first image, each element of the feature vector has a 50% probability of being randomly set to zero. For the second, noise is

added to each component based on the standard deviation of the projected image. For the third, five images from the same class as the original image are selected, and their PCA vectors are computed. With a 5% probability, components from the original PCA vector are swapped with corresponding components from the other five PCA vectors. The three perturbed PCA vectors are then transformed back using the inverse PCA transform to produce the augmented images.

- APP5: Similar to APP4, this augmentation generates three new images using the perturbation method described above. However, instead of using PCA, the Discrete Cosine Transform (DCT) is applied. It should be noted that the DC coefficient is never changed during this transformation.
- APP6: This augmentation is designed specifically for color images. It creates three new images by color shifting and by altering contrast and sharpness. Contrast alteration is achieved by linearly scaling the original image's contrast between the lowest value (a) and the highest value (b) allowed for the augmented image. Any pixel in the original image outside this range is mapped to 0 if it is lower than a or 255 if it is greater than b. Sharpness is modified by blurring the original image with a Gaussian filter (variance = 1) and subtracting the blurred image from the original. Color shifting is performed by applying integer shifts to the three RGB filters, and each shift is added to one of the three channels in the original image.

These data augmentation techniques aim to increase the diversity of the training data, helping the networks learn robust features and improve their performance on the classification task.

3.3. Parameter Ensembling via Perturbation (PEP)

Another approach for creating ensembles is PEP [59]. In this method, only a single network is trained, but the ensemble is formed by introducing perturbations to the weights of the final network using additive Gaussian random noise. The researchers who designed this method have demonstrated that by appropriately adjusting the amount of noise the performance of the original network can be surpassed. The determination of the optimal noise level can be accomplished, as in [59], through experimentation on a validation set.

In our evaluation, we have tested several variants of PEP, which include the original version and the following new ones proposed here:

- Dout: similar to drop-out: 2% of the weights zeroed out;
- DCTa: each set of weights is projected onto a Discrete Cosine Transform (DCT) space, with (3.33%) randomly chosen DCT coefficients set to zero (the DC component is never zeroed out), after which the inverse DCT is applied.
- DCTb: each set of weights is projected onto a Discrete Cosine Transform (DCT) space where a small amount of random noise is injected (the DC component is never perturbed), after which the inverse DCT is applied.
- PEPa: method similar to the original version, but where a small amount of random noise is injected.
- PEPb: the same idea as PEPa, but noise is injected in a different manner.

We apply these methods as follows. First, we train the network for 20 epochs to obtain netA. Next, we apply weight perturbation on netA, then train the network again for a single epoch. The resulting network is netP. Perturbation is performed five times, and each time the perturbation is applied to netA. In this way, we obtain five netPs (netP(1), netP(2), ..., netP(5)). The final output is given by the average rule between the output of netA and the five netPs.

Below is the pseudo-code of the proposed approaches.

Listing 1: Dout: similar to dropout filter

```
Perturbation = rand(size(Weights));
% Weights are the weights of the given net
% Perturbation is a tensor of the same size as the
% set of weights of the net, randomly initialized to [0,1]
Perturbation = Perturbation < 0.98;  % 2% of the values are set to zero
Weights = Weights.*Perturbation;     % some weights are zeroed out
```

Listing 2: DCTa: DCT based perturbation approach

```
    for each layer
        for each channel
            IMG = Weights(layer,channel);
            % weights of a given channel-layer are stored
            dctProj = dct2(IMG);  % DCT projection
            dctProj_reset = dctProj;
            % reset some random dct coefficients
            dctProj_reset("random indexes") = 0;
            % DC component is never zeroed out
            dctProj_reset(1,1) = dctProj(1,1);
            Weights(layer,channel) = idct2(dctProj_reset); % retroprojection
        end
    end
```

Listing 3: DCTb: DCT based perturbation approach

```
for each layer
    for each channel
        IMG = Weights(layer,channel);
        % weights of a given channel-layer are stored
        dctProj = dct2(IMG);  % DCT projection
        % standard deviation of the values of the weights
        noise = std(dctProj) / 4;
        % random noise
        dctProjNew = dctProj + (rand-0.5) .* noise;
        % rand is random between 0 and 1
        dctProjNew(1,1) = dctProj(1,1); % DC component is never zeroed out
        Weights(layer, channel) = idct2(dctProjNew); % retroprojection
    end
end
```

Listing 4: PEPa: Method 3 is similar to DCT1

```
sigma = 0.002;
Weights = Weights + rand(size(Weights)) .* sigma;
% Weights are the weights of the given net
```

Listing 5: PEPb: Method 3 is similar to DCT1

```
sigma = 0.2;
Weights = Weights .* (1 + rand(size(Weights)) .* sigma);
% Weights are the weights of the given net
```

4. Experimental Results

In this section, we detail the experimental analysis of the ensemble methods.

4.1. Datasets

In our study, we utilized the following data sets to evaluate the performance of our approach:

- HE (2D HeLa data set [60]): This data set has 862 fluorescence microscopy images of HeLa cells stained with different fluorescent dyes specific to various organelles. The data set is well-balanced and divided into ten classes representing different organelles, including DNA (Nuclei), ER (Endoplasmic reticulum), Giantin (cis/medial Golgi), GPP130 (cis Golgi), Lamp2 (Lysosomes), Nucleolin (Nucleoli), Actin, TfR (Endosomes), Mitochondria, and Tubulin. A 5-fold cross-validation is applied.
- MA (C. elegans Muscle Age data set [61]): This data set focuses on classifying the age of C. elegans nematodes. It has 257 images of C. elegans muscles collected at four different ages, representing distinct classes based on age. A 5-fold cross-validation is applied.
- BG (Breast Grading Carcinoma [62]): This data set, obtained from Zenodo (record: 834910#.Wp1bQ-jOWUl), has 300 annotated histological images of breast tissues from patients diagnosed with invasive ductal carcinoma. The data set is categorized into three classes representing different grades (1-3) of carcinoma. A 5-fold cross-validation is applied.
- LAR (Laryngeal data-set [63]): Obtained from Zenodo (record: 1003200#.WdeQcnBx0nQ), has 1320 images of laryngeal tissues. It includes both healthy and early-stage cancerous tissues, representing a total of four tissue classes. This data set is split into three folds by the original authors.
- POR (portrait dataset) data set [64] focuses specifically on portrait images of humans. It is designed to evaluate segmentation performance in the context of portrait photography, considering factors such as facial features, skin tones, and background elements. This dataset includes 1447 images for training and 289 images for validation. POR can be accessed at https://github.com/HYOJINPARK/ExtPortraitSeg.

By utilizing these diverse data sets, we aimed to evaluate the performance of our ensembling approaches across various imaging tasks and scenarios (mostly medical), providing a comprehensive analysis of the different methods' effectiveness on small data sets.

4.2. Results

In the first test reported in Table 1, we compare the performance of RE and SE both with APP3 data augmentation (DA) and without a data augmentation step (noDA). We do this by varying the number of classifiers that build the ensemble. In this table, SE(x) means that we combine by sum rule 'x' networks coupled with the stochastic approach for replacing the activation function layers, and RE(x) means that we combine by sum rule 'x' standard ResNet50, where each network is simply re-trained on the training set.

Table 1. Performance on the different datasets

	HE	MA	BG	LAR	POR	Average
RE(1)-noDA	94.65	92.50	91.67	90.98	85.74	91.11
RE(14)-noDA	96.05	95.00	90.33	94.02	87.15	92.51
RE(30)-noDA	95.81	94.58	90.67	94.02	87.15	92.44
RE(1)-DA	95.93	95.83	92.67	94.77	86.29	93.10
RE(14)-DA	**96.63**	97.50	94.33	95.76	88.24	94.49
RE(30)-DA	96.33	**98.33**	94.00	95.83	88.56	94.61
SE(14)-noDA	95.47	95.42	92.67	94.62	88.02	93.24
SE(30)-noDA	95.58	96.25	92.67	95.00	88.77	93.65
SE(14)-DA	**96.63**	**98.33**	94.67	95.98	88.67	94.86
SE(30)-DA	96.33	**98.33**	**95.00**	**96.21**	**89.00**	**94.97**

From the results reported in Table 1, we can draw the following conclusions:

- Data augmentation is useful both for RE and SE.
- Both RE and SE increase performance only slightly from x = 14 to x = 30.
- SE outperforms RE.

In Table 2, we compare the different data augmentation approaches. In this case, there is no clear winner. In each data set, the rank position of the single data augmentation method varies.

Table 2. Data augmentation approaches.

DataAUG	HE	MA	BG	LAR	POR	Average
DA1	95.12	95.00	93.00	92.95	87.05	92.62
DA2	96.63	95.83	94.00	95.08	85.97	93.50
DA3	95.93	95.83	92.67	94.77	86.29	93.10
DA4	95.23	93.33	92.33	94.62	84.90	92.08
DA5	95.35	91.25	91.33	95.45	86.41	91.95
DA6	92.44	91.25	92.33	94.39	87.37	91.55
ALL	**96.74**	**97.50**	**94.00**	**96.06**	**89.00**	**94.66**
RE(6)-DA	96.40	97.08	93.67	95.98	88.45	94.31

Examining Table 3, no clear winner is evident, even when we compare the different PEP-based approaches. However, in BG and LAR the new proposed DCTa outperforms both PEPa and PEPb. The most interesting result is that ALL (the fusion of the five PEP-based approaches) outperforms PEPa(5), a set of five PEPa methods (each obtained by retraining the networks). PEPa(5) always performs worse than ALL. Interestingly, ALL almost always improves RE(5)-noDA, implying that PEP-based methods are indeed an effective way to build network ensembles. The results of this experiment show that it is useful to apply different perturbation approaches to obtain a set of networks. It should be noted that for the sake of computation time we did not use DA in this test.

Table 3. PEP variants.

	HE	MA	BG	LAR	POR	Average
DropOut	94.53	95.00	88.33	92.65	84.57	91.10
DCTa	94.88	93.33	**92.00**	94.09	85.11	91.88
DCTb	93.95	94.17	90.00	92.35	84.79	91.05
PEPa	95.58	93.33	89.67	92.20	85.22	91.20
PEPb	94.77	92.08	89.33	92.58	85.11	90.77
PEPa(5)	95.93	96.25	90.33	94.02	86.94	92.69
ALL	**96.05**	**97.08**	90.67	**94.24**	**86.95**	**93.00**
RE(5)-noDA	95.47	94.58	91.33	93.48	86.82	92.33

In Tables 4 and 5, we compare different approaches for building an ensemble of ResNet50s. The methods are those explained above. We also report in this table the performance of:

- StocDA_PEP(18), which was created as follows: for each DA method, we train three networks for a total of eighteen. Each network is then coupled with one of the five PEP variants (randomly chosen).
- StocDA(18), which was created as follows: For each DA method, we train three SE networks for a total of eighteen.

Table 4. Comparison among ensembles: accuracy.

	HE	MA	BG	LAR	POR	PEST	InfLAR	TRIZ	Average
RE(1)-DA	95.93	95.83	92.67	94.77	86.29	93.70	95.56	98.78	94.19
RE(18)-DA	96.33	**98.33**	94.33	95.61	88.13	93.87	96.30	98.78	95.21
SE(18)-DA	**96.51**	**98.33**	**95.00**	96.06	88.56	94.36	96.67	98.95	95.55
StocDA(18)	96.10	96.67	94.33	96.81	89.96	**94.48**	96.53	98.95	95.47
StocDA_PEP(18)	96.40	97.50	94.00	**96.82**	**91.68**	94.14	**97.08**	**99.13**	**95.84**

Table 5. Comparison among ensembles: EUC.

	HE	MA	BG	LAR	POR	PEST	InfLAR	TRIZ	Average
RE(1)-DA	0.40	0.79	2.74	0.41	2.69	0.75	0.54	0.10	1.05
RE(18)-DA	0.22	0.16	2.32	0.18	2.05	0.71	0.49	0.13	0.78
SE(18)-DA	0.14	**0.06**	2.72	0.14	1.88	0.57	0.49	0.05	0.75
StocDA(18)	0.15	0.10	2.96	0.09	1.36	0.53	0.41	0.04	0.70
StocDA_PEP(18)	**0.10**	0.07	**1.67**	**0.07**	**1.31**	**0.52**	**0.40**	**0.03**	**0.52**

In Table 4 the accuracy obtained by the compared methods is reported, instead, in table 5 the error under the ROC curve (EUC) is reported, it is defined as (100-"Area under the ROC curve)%. In addition, we use three other datasets in the following tables for a more robust statistical comparison:

- PEST, [65], It is a dataset of 563 pest images, 10 classes, commonly found on plants. We use the split training-test sets suggested by the original authors.
- InfLAR, [66], it is a dataset of 720 images, four classes, extracted from laryngoscopic videos. We use the split training-test sets (three different folds) suggested by the original authors.
- TRIZ, [67], it is a dataset of 574 gastric lesion type images, four classes; as suggested by the original authors we apply a 10-fold cross-validation.

From Tables 4 and 5, we can draw the following conclusions:

- In data sets with gray-level images, StocDA_PEP(18) and StocDA(18) are less useful (probably because some data augmentation methods are suitable for color images), whereas in LAR, POR, PEST, InfLAR, and TRIZ they fare better;
- There is a noticeable difference in performance between each ensemble compared to the original single ResNet50.
- considering the well-known Wilcoxon-signed rank test, considering both the performance indicators (i.e. accuracy and EUC): RE(18)-DA outperforms RE(1)-DA with a p-value of 0.01; SE(18)-DA outperforms RE(18)-DA with a p-value of 0.05; StocDA(18) obtains similar performance with respect to SE(18)-DA (due to the gray level images datasets); StocDA_PEP(18) outperforms StocDA(18) with a p-value of 0.05.

Finally, in Table 6, we report tests on computation time.

Table 6. Inference time of a batch size of 100 images

GPU	GPU Year	Single ResNet50	Ensemble 15 ResNet50
GTX 1080	2016	0.36 sec	5.58 sec
Titan Xp	2017	0.31 sec	4.12 sec
Titan RTX	2018	0.22 sec	2.71 sec
Titan V100	2018	0.20 sec	2.42 sec

The hardware improvements, as clearly expected, reduce the inference time; there are several applications where it is not a problem to classify 100 images in just a few seconds.

5. Conclusion

The aim of this study was to evaluate the effectiveness of advanced ensemble deep learning techniques. Various methods for creating ensembles, specifically focused on CNNs, were investigated. The main intention of this work was on comparing the performance of standalone ensembles and various combinations of the following ensembling techniques: replacing ReLU layers with different activation functions, employing various data augmentation techniques, and utilizing several methods to perturb the network weights. To assess the performance of these ensembles, the well-known ResNet50 model was employed due to its balanced tradeoff between performance and training time. The evaluation was carried out on five challenging image datasets encompassing diverse tasks.

The experimental results demonstrated that the proposed ensemble of CNNs outperformed standalone methods for building CNN ensembles. However, additional research is necessary to investigate the performance benefits of this approach on a broader range of data sets, such as Computer Tomography (CT), Magnetic Resonance Imaging (MRI), and image/tumor segmentation. Experiments using different network topologies also need to be performed.

Conducting such investigations poses challenges, however, due to the substantial computational resources required for CNN analysis. Nevertheless, these studies are vital for improving the accuracy of deep learning systems in image and data classification tasks.

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

Funding: This research received no external funding.

Data Availability Statement: All the resources required to replicate our experiments are available at https://github.com/LorisNanni.

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

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

References

1. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* **2012**, *25*.
2. Haggenmüller, S.; Maron, R.C.; Hekler, A.; Utikal, J.S.; Barata, C.; Barnhill, R.L.; Beltraminelli, H.; Berking, C.; Betz-Stablein, B.; Blum, A.; others. Skin cancer classification via convolutional neural networks: systematic review of studies involving human experts. *European Journal of Cancer* **2021**, *156*, 202–216.
3. 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.
4. Horie, Y.; Yoshio, T.; Aoyama, K.; Yoshimizu, S.; Horiuchi, Y.; Ishiyama, A.; Hirasawa, T.; Tsuchida, T.; Ozawa, T.; Ishihara, S.; others. Diagnostic outcomes of esophageal cancer by artificial intelligence using convolutional neural networks. *Gastrointestinal endoscopy* **2019**, *89*, 25–32.
5. Qummar, S.; Khan, F.G.; Shah, S.; Khan, A.; Shamshirband, S.; Rehman, Z.U.; Khan, I.A.; Jadoon, W. A deep learning ensemble approach for diabetic retinopathy detection. *Ieee Access* **2019**, *7*, 150530–150539.
6. Pan, D.; Zeng, A.; Jia, L.; Huang, Y.; Frizzell, T.; Song, X. Early detection of Alzheimer's disease using magnetic resonance imaging: a novel approach combining convolutional neural networks and ensemble learning. *Frontiers in neuroscience* **2020**, *14*, 259.
7. Nanni, L.; Loreggia, A.; Lumini, A.; Dorizza, A. A Standardized Approach for Skin Detection: Analysis of the Literature and Case Studies. *Journal of Imaging* **2023**, *9*, 35.
8. Thanapol, P.; Lavangnananda, K.; Bouvry, P.; Pinel, F.; Leprévost, F. Reducing overfitting and improving generalization in training convolutional neural network (CNN) under limited sample sizes in image recognition. 2020-5th International Conference on Information Technology (InCIT). IEEE, 2020, pp. 300–305.

9. Campagner, A.; Ciucci, D.; Svensson, C.M.; Figge, M.T.; Cabitza, F. Ground truthing from multi-rater labeling with three-way decision and possibility theory. *Information Sciences* **2021**, *545*, 771–790.
10. Panch, T.; Mattie, H.; Celi, L.A. The "inconvenient truth" about AI in healthcare. *NPJ digital medicine* **2019**, *2*, 77.
11. Bravin, R.; Nanni, L.; Loreggia, A.; Brahnam, S.; Paci, M. Varied Image Data Augmentation Methods for Building Ensemble. *IEEE Access* **2023**, *11*, 8810–8823.
12. Nanni, L.; Cuza, D.; Lumini, A.; Loreggia, A.; Brahman, S. Polyp Segmentation with Deep Ensembles and Data Augmentation. In *Artificial Intelligence and Machine Learning for Healthcare: Vol. 1: Image and Data Analytics*; Springer, 2022; pp. 133–153.
13. Nanni, L.; Fantozzi, C.; Loreggia, A.; Lumini, A. Ensembles of Convolutional Neural Networks and Transformers for Polyp Segmentation. *Sensors* **2023**, *23*, 4688.
14. Nanni, L.; Lumini, A.; Loreggia, A.; Brahnam, S.; Cuza, D. Deep ensembles and data augmentation for semantic segmentation. In *Diagnostic Biomedical Signal and Image Processing Applications with Deep Learning Methods*; Elsevier, 2023; pp. 215–234.
15. Papanastasopoulos, Z.; Samala, R.K.; Chan, H.P.; Hadjiiski, L.; Paramagul, C.; Helvie, M.A.; Neal, C.H. Explainable AI for medical imaging: deep-learning CNN ensemble for classification of estrogen receptor status from breast MRI. Medical imaging 2020: Computer-aided diagnosis. SPIE, 2020, Vol. 11314, pp. 228–235.
16. He, X.; Zhou, Y.; Wang, B.; Cui, S.; Shao, L. Dme-net: Diabetic macular edema grading by auxiliary task learning. International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2019, pp. 788–796.
17. Coupé, P.; Mansencal, B.; Clément, M.; Giraud, R.; de Senneville, B.D.; Ta, V.T.; Lepetit, V.; Manjon, J.V. AssemblyNet: A large ensemble of CNNs for 3D whole brain MRI segmentation. *NeuroImage* **2020**, *219*, 117026.
18. Savelli, B.; Bria, A.; Molinara, M.; Marrocco, C.; Tortorella, F. A multi-context CNN ensemble for small lesion detection. *Artificial Intelligence in Medicine* **2020**, *103*, 101749.
19. Cornelio, C.; Donini, M.; Loreggia, A.; Pini, M.S.; Rossi, F. Voting with random classifiers (VORACE): theoretical and experimental analysis. *Auton. Agent* **2021**, *35*, 2. doi:10.1007/s10458-021-09504-y.
20. Yao, X.; Liu, Y. Making use of population information in evolutionary artificial neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **1998**, *28*, 417–425.
21. Opitz, D.; Shavlik, J. Generating accurate and diverse members of a neural-network ensemble. *Advances in neural information processing systems* **1995**, *8*.
22. Liu, Y.; Yao, X.; Higuchi, T. Evolutionary ensembles with negative correlation learning. *IEEE Transactions on Evolutionary Computation* **2000**, *4*, 380–387.
23. Rosen, B.E. Ensemble learning using decorrelated neural networks. *Connection science* **1996**, *8*, 373–384.
24. Liu, Y.; Yao, X. Ensemble learning via negative correlation. *Neural networks* **1999**, *12*, 1399–1404.
25. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778. doi:10.1109/CVPR.2016.90.
26. Kassani, S.H.; Kassani, P.H.; Wesolowski, M.J.; Schneider, K.A.; Deters, R. Classification of histopathological biopsy images using ensemble of deep learning networks. *arXiv preprint arXiv:1909.11870* **2019**.
27. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* **2014**.
28. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4510–4520.
29. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.
30. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, 26 June–1 July 2016, pp. 770–778.
31. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.

32. Chollet, F. Xception: Deep learning with depthwise separable convolutions. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1251–1258.
33. Liu, K.; Zhang, M.; Pan, Z. Facial expression recognition with CNN ensemble. 2016 international conference on cyberworlds (CW). IEEE, 2016, pp. 163–166.
34. Goodfellow, I.J.; Erhan, D.; Carrier, P.L.; Courville, A.; Mirza, M.; Hamner, B.; Cukierski, W.; Tang, Y.; Thaler, D.; Lee, D.H.; others. Challenges in representation learning: A report on three machine learning contests. Neural Information Processing: 20th International Conference, ICONIP 2013, Daegu, Korea, November 3-7, 2013. Proceedings, Part III 20. Springer, 2013, pp. 117–124.
35. Kumar, A.; Kim, J.; Lyndon, D.; Fulham, M.; Feng, D. An ensemble of fine-tuned convolutional neural networks for medical image classification. *IEEE journal of biomedical and health informatics* **2016**, *21*, 31–40.
36. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.
37. Gilbert, A.; Piras, L.; Wang, J.; Yan, F.; Ramisa, A.; Dellandrea, E.; Gaizauskas, R.J.; Villegas, M.; Mikolajczyk, K.; others. Overview of the ImageCLEF 2016 Scalable Concept Image Annotation Task. CLEF (Working Notes), 2016, pp. 254–278.
38. Pandey, P.; Deepthi, A.; Mandal, B.; Puhan, N.B. FoodNet: Recognizing foods using ensemble of deep networks. *IEEE Signal Processing Letters* **2017**, *24*, 1758–1762.
39. Breiman, L. Bagging predictors. *Machine learning* **1996**, *24*, 123–140.
40. Wolpert, D.H.; Macready, W.G. An efficient method to estimate bagging's generalization error. *Machine Learning* **1999**, *35*, 41–55.
41. Bauer, E.; Kohavi, R. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning* **1999**, *36*, 105–139.
42. Kim, P.K.; Lim, K.T. Vehicle type classification using bagging and convolutional neural network on multi view surveillance image. Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2017, pp. 41–46.
43. Dong, X.; Qian, L.; Huang, L. A CNN based bagging learning approach to short-term load forecasting in smart grid. 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). IEEE, 2017, pp. 1–6.
44. Guo, J.; Gould, S. Deep CNN ensemble with data augmentation for object detection. *arXiv preprint arXiv:1506.07224* **2015**.
45. Gan, Y.; Chen, J.; Xu, L. Facial expression recognition boosted by soft label with a diverse ensemble. *Pattern Recognition Letters* **2019**, *125*, 105–112.
46. Antipov, G.; Berrani, S.A.; Dugelay, J.L. Minimalistic CNN-based ensemble model for gender prediction from face images. *Pattern recognition letters* **2016**, *70*, 59–65.
47. Nanni, L.; Brahnam, S.; Paci, M.; Ghidoni, S. Comparison of different convolutional neural network activation functions and methods for building ensembles for small to midsize medical data sets. *Sensors* **2022**, *22*, 6129.
48. Nanni, L.; Lumini, A.; Ghidoni, S.; Maguolo, G. Stochastic selection of activation layers for convolutional neural networks. *Sensors* **2020**, *20*, 6. doi:10.3390/s20061626.
49. Berno, F.; Nanni, L.; Maguolo, G.; Brahnam, S. Ensembles of convolutional neural networks with different activation functions for small to medium size biomedical datasets. *Machine Learning in Medicine; CRC Press Taylor & Francis Group: Boca Raton, FL, USA* **2021**.
50. Ju, C.; Bibaut, A.; van der Laan, M. The relative performance of ensemble methods with deep convolutional neural networks for image classification. *Journal of Applied Statistics* **2018**, *45*, 2800–2818.
51. Harangi, B. Skin lesion classification with ensembles of deep convolutional neural networks. *Journal of biomedical informatics* **2018**, *86*, 25–32.
52. Lyksborg, M.; Puonti, O.; Agn, M.; Larsen, R. An ensemble of 2D convolutional neural networks for tumor segmentation. Image Analysis: 19th Scandinavian Conference, SCIA 2015, Copenhagen, Denmark, June 15-17, 2015. Proceedings 19. Springer, 2015, pp. 201–211.
53. Minetto, R.; Segundo, M.P.; Sarkar, S. Hydra: An ensemble of convolutional neural networks for geospatial land classification. *IEEE Transactions on Geoscience and Remote Sensing* **2019**, *57*, 6530–6541.

54. Dong, X.; Yu, Z.; Cao, W.; Shi, Y.; Ma, Q. A survey on ensemble learning. *Frontiers of Computer Science* **2020**, *14*, 241–258.
55. Brown, G.; Wyatt, J.; Harris, R.; Yao, X. Diversity creation methods: a survey and categorisation. *Information fusion* **2005**, *6*, 5–20.
56. Sagi, O.; Rokach, L. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **2018**, *8*, e1249.
57. Lumini, A.; Nanni, L.; Maguolo, G. Deep learning for plankton and coral classification. *Applied Computing and Informatics* **2020**.
58. Duch, W.; Jankowski, N. Survey of neural transfer functions. *Neural computing surveys* **1999**, *2*, 163–212.
59. Mehrtash, A.; Abolmaesumi, P.; Golland, P.; Kapur, T.; Wassermann, D.; Wells, W. Pep: Parameter ensembling by perturbation. *Advances in neural information processing systems* **2020**, *33*, 8895–8906.
60. 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.
61. Shamir, L.; Orlov, N.; Mark Eckley, D.; Macura, T.J.; Goldberg, I.G. IICBU 2008: a proposed benchmark suite for biological image analysis. *Medical & biological engineering & computing* **2008**, *46*, 943–947.
62. 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*, e0185110.
63. 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. *Journal of Medical Imaging* **2017**, *4*, 034502–034502.
64. Kim, Y.W.; Byun, Y.C.; Krishna, A.V.N. Portrait Segmentation Using Ensemble of Heterogeneous Deep-Learning Models. *Entropy* **2021**, *23*. doi:10.3390/e23020197.
65. Deng, L.; Wang, Y.; Han, Z.; Yu, R. Research on insect pest image detection and recognition based on bio-inspired methods. *Biosystems Engineering* **2018**, *169*, 139–148, [1910.00296]. doi:10.1016/J.BIOSYSTEMSENG.2018.02.008.
66. Patrini, I.; Ruperti, M.; Moccia, S.; Mattos, L.S.; Frontoni, E.; De Momi, E. Transfer learning for informative-frame selection in laryngoscopic videos through learned features. *Medical and Biological Engineering and Computing* **2020**, *58*, 1225–1238. doi:10.1007/S11517-020-02127-7/TABLES/8.
67. Zhao, R.; Zhang, R.; Tang, T.; Feng, X.; Li, J.; Liu, Y.; Zhu, R.; Wang, G.; Li, K.; Zhou, W.; Yang, Y.; Wang, Y.; Ba, Y.; Zhang, J.; Liu, Y.; Zhou, F. TriZ-a rotation-tolerant image feature and its application in endoscope-based disease diagnosis. *Computers in Biology and Medicine* **2018**, *99*, 182–190. doi:10.1016/J.COMPBIOMED.2018.06.006.

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