electronics

MDPI

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, 35122 Padova, Italy
[2] Department of Information Engineering, University of Brescia, 25123 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 pretrained 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 datasets representing different tasks demonstrate that our proposed methods for building deep ensembles produces superior results.

Keywords: convolutional neural networks; ensembles; fusion

Citation: Nanni, L.; Loreggia, A.; Brahnam, S. Comparison of Different Methods for Building Ensembles of Convolutional Neural Networks. *Electronics* **2023**, *12*, 4428. https://doi.org/10.3390/electronics12214428

Academic Editors: Jungpil Shin, Md. Al Mehedi Hasan and Hoang D. Le

Received: 1 September 2023
Revised: 20 October 2023
Accepted: 24 October 2023
Published: 27 October 2023

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 (e.g., [1,2]), medical (e.g., [3,4]), and engineering domains (e.g., [5,6]). 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 [7]. Not only do CNNs surpass traditional classifiers, but they also often outperform the recognition abilities of human beings. In the medical realm, CNNs have demonstrated superior performance compared to human experts in tasks such as skin cancer detection [8,9], identification of skin lesions on the face and scalp, and diagnosis of esophageal cancer (e.g., [10]). These remarkable achievements have naturally triggered a substantial increase in research focused on utilizing CNNs and other deep-learning techniques in medical imaging.

For instance, deep-learning models have emerged as the state-of-the-art for diagnosing conditions like diabetic retinopathy [11], Alzheimer's disease [12], skin detection [13], gastrointestinal ulcers, and various types of cancer, as demonstrated in recent reviews and studies (see, for instance, [14,15]). Enhancing performance within the medical field carries the greater real impact of this technology compared to other applications.

CNNs, however, have limitations. It is widely recognized that they require many samples to avoid overfitting [16]. Acquiring image collections numbering in the hundreds

of thousands for proper CNN training is an enormous enterprise [17]. In certain medical domains, it is prohibitively labor-intensive and costly [18]. Several well-established techniques have been developed to address the issue of overfitting with limited data, the two most common being transfer learning using pretrained CNNs and data augmentation [19,20]. The literature is abundant with studies investigating both methods, and it has been observed that combining the two yields better results (e.g., [21]).

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 pretrained CNNs [22]. Ensemble learning is a powerful technique in machine learning that aims to enhance predictive performance by combining the outputs of multiple classifiers [23]. The fundamental idea behind ensemble learning is to introduce diversity among the individual classifiers so that they collectively provide more accurate and robust predictions. This diversification can be achieved through various means, each contributing to the ensemble's overall effectiveness [24].

One common approach to creating diversity among classifiers is to train each classifier on different subsets or variations of the available data [25]. This approach, known as data sampling or bootstrapping, allows each classifier to focus on different aspects or nuances within the dataset, which can lead to improved generalization and robustness. Another technique for introducing diversity is to use different types of CNN architectures within the ensemble [26]. By combining CNNs with distinct architectural features, such as varying kernel sizes, filter depths, or connectivity patterns, the ensemble can capture different aspects of the underlying data distribution, enhancing its overall predictive power [27,28]. In addition to varying architectural aspects, ensemble diversity can also be achieved by modifying network depth. Some classifiers within the ensemble may have shallower network architectures, while others may be deeper. This diversity in network depth can help the ensemble address different levels of complexity within the data, improving its adaptability to varying patterns and structures.

Furthermore, the ensemble can introduce diversity by using different activation functions within the neural networks. Activation functions play a crucial role in determining how information flows through the neural network layers. By employing a variety of activation functions, the ensemble can capture different types of non-linear relationships in the data, enhancing its ability to model complex patterns. Figure 1 depicts an example of a neural network in which each layer adopts an activation function that could be chosen at random among a set of available ones. The chosen activation function is then used by all the neurons in that layer.

Many robust CNN ensembles have been reported in recent years, showcasing their effectiveness in various applications. For instance, in [29], a deep CNN ensemble was developed for classifying ER (Estrogen Receptor) status from DCE-MRI (Dynamic Contrast-Enhanced Magnetic Resonance Imaging) breast volumes. In [30], the authors focused on diabetic muscular edema diagnosis and employed a hierarchical ensemble approach. In [31], a CNN ensemble was designed for whole-brain segmentation, while in [32] an ensemble approach for small lesion detection was proposed. In each of these cases, the ensemble was shown to perform better than standalone CNNs.

This research paper focuses on image classification using ensembles of CNNs. In particular, we focus on two models as baselines for our experimental part: ResNet50 [33] and MobileNetV2 [34]. ResNet50 has been chosen because of its balanced tradeoff between performance and training time. MobileNetV2 MobileNetV2 has been chosen to test whether the proposal can be adopted also in those scenarios with poor computational power, such as edge computing.

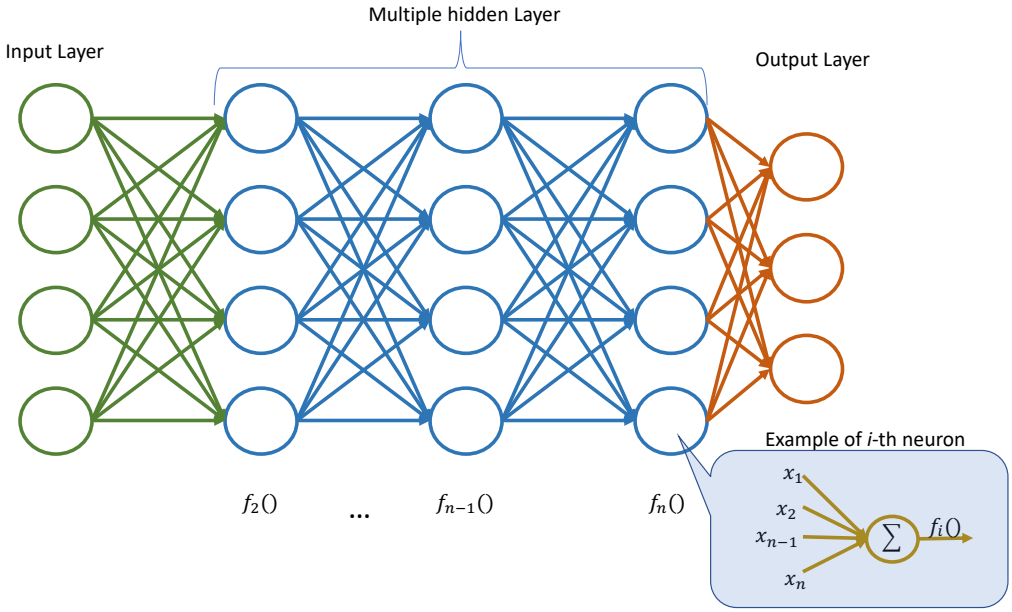

Figure 1. Example of neural network with multiple hidden layers. Each layer adopts a (possibly) different activation function to be used by all the neurons in that layer.

The goal of this work is to perform an exhaustive evaluation across many datasets 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. The first method changes the way neurons in the model compute the output. This allows us to represent complex patterns and relationships within the dataset. Different activation functions have various characteristics, affecting how neurons respond to their inputs and impacting the network's ability to capture intricate data patterns (this will be discussed in Section 3.1). The second method generates different datasets by applying various transformations or modifications to the existing data. In such a way, each model is trained on a different set of data and thus a different exploration of the hypothesis space (this will be discussed in Section 3.3). The third method generates models that differ to one another in their configuration. This provides models that are potentially able to get out of local minima (this will be discussed in Section 3.4).

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 several different datasets;
- 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 pretrained CNNs, employing various training algorithms, and applying distinct rules for combining networks.

The most intuitive approach to forming an ensemble involves training different models and then combining their outputs. Identifying the optimal classifier for a complex task can be a challenging endeavor [23]. Various classifiers may excel in leveraging the distinctive characteristics of specific areas within the given domain, potentially resulting in higher accuracy exclusively within those particular regions [35,36].

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 [37], Kassani et al. employed an ensemble of VGG19 [38], MobileNet [34], and DenseNet [39] to classify histopathological biopsies, showing that the ensemble consistently achieved better performance than each individual network across four different datasets. Similarly, Qummar et al. [11] proposed an ensemble comprising ResNet50 [33], Inception v3 [40], Xception [41], DenseNet121, and DenseNet169 [39] to detect diabetic retinopathy.

In their study, Liu et al. [42] 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 [43]. Similarly, Kumar et al. [44] introduced an ensemble of pretrained AlexNet and GoogleNet [45] models from ImageNet, which were then fine-tuned on the ImageCLEF 2016 collection dataset [46]. 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. [47] 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 to generating independent classifiers [23]. This can be achieved through various methods, with one classic technique being bagging [48–50]. 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. Figure 2 depicts the bagging process. Examples of this approach to building ensembles include the work of Kim and Lim [51], who proposed a bagging-based approach to train three distinct CNNs for vehicle-type classification. Similarly, Dong et al. [52] applied bagging and CNNs to improve short-term load forecasting in a smart grid, resulting in a significant reduction of the mean absolute percentage error (MAPE) from 33.47 to 28.51. As another example, Guo and Gould [53] 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 models at the end of the process. We can enhance the diversity among the final models in many ways: for instance, by employing different initialization of the initial model, or by adopting different optimization algorithms or loss functions during the training phase. Figure 3 depicts an example of an ensemble which adopt different optimization algorithms to introduce diversity. For instance, the authors in [54] constructed an ensemble for facial expression recognition using soft-label perturbation, where different losses were propagated for different samples. Similarly, Antipov et al. [55] utilized different network initializations to train multiple networks for gender predictions from face images.

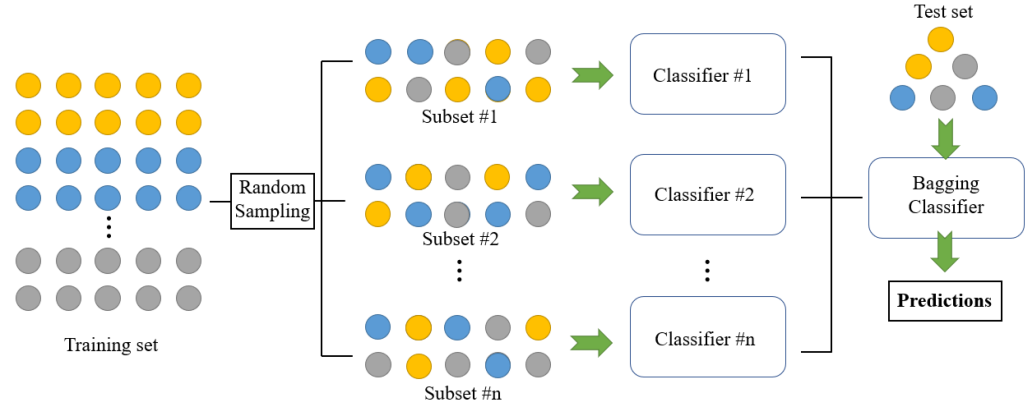

Figure 2. Structure of bagging classifier [56].

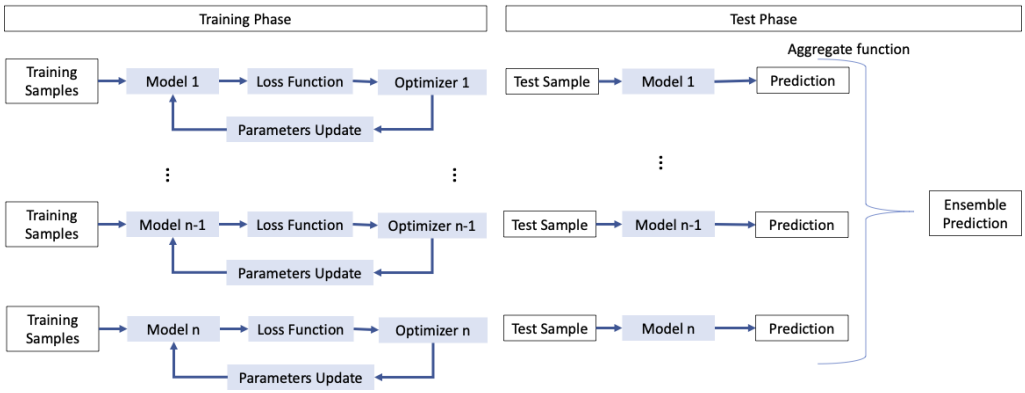

Figure 3. Example of ensemble that can use different optimization algorithms to introduce diversity.

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 [57]. One way to implement the latter approach is to select a random activation function from a pool for each layer in the original network [57]. The diversity introduced in this way makes activation functions an excellent candidate for generating ensembles of deep learners 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 [58–61]. Another common technique frequently cited in the literature is to average the softmax outputs of the networks [62–64].

3. Methods

The ensembles in this study are constructed using sets of ResNet50 [33] (short for Residual Network with 50 layers or using MobileNetV2 [34]. ResNet50 is a variant of the original ResNet architecture, known for its deep structure, which mitigates the vanishing gradient problem by introducing skip connections or residual blocks. These residual blocks allow for the training of very deep neural networks effectively. ResNet-50 specifically consists of 50 convolutional layers, including residual blocks, and is pretrained on a large dataset like ImageNet, making it capable of extracting high-level features from images. It is a popular choice for various image-related tasks, including object detection, image recognition, and transfer learning. 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. Figure 4 presents a high-level visualization of the network's architecture.

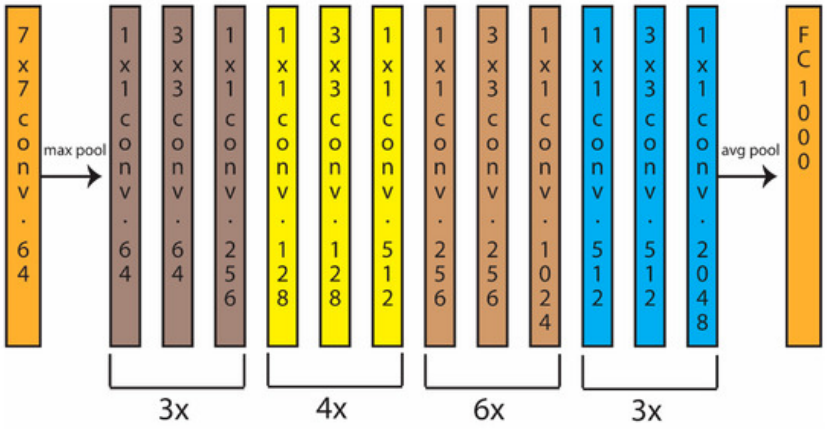

Figure 4. ResNet50 architecture.

MobileNetV2 is a lightweight deep neural network architecture designed for efficient and high-performance image classification and object-detection tasks on mobile and embedded devices. It is an evolution of the original MobileNet architecture, developed by Google, with a focus on achieving better accuracy and efficiency. MobileNetV2 is characterized by its use of depthwise separable convolutions, which reduce computational complexity while preserving representational capacity. This architectural choice allows MobileNetV2 to achieve a good balance between model size, inference speed, and accuracy. It is especially well suited for applications where resource constraints, such as limited computational power and memory, are a consideration. Some key features and advantages of MobileNetV2 include its ability to efficiently handle a wide range of image sizes, making it adaptable for various tasks. It also provides flexibility in terms of model size, allowing users to choose from a range of predefined configurations to meet their specific requirements. MobileNetV2 has been widely adopted in mobile and edge computing applications, including real-time object detection, image classification, and even on-device machine learning tasks. Its efficiency and versatility make it a valuable tool for developers looking to deploy deep-learning models on resource-constrained devices.

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.

3.1. Activation Functions

An activation function in machine learning is a mathematical function that introduces non-linearity into artificial neural networks, enabling them to model complex patterns and make accurate predictions. In this research, we explore a comprehensive set of over twenty activation functions for constructing CNN ensembles. Among these activation functions are widely recognized ones like ReLU, Leaky ReLU, ELU, SELU, PReLU, APLU, SReLU, MeLU, Splash, Mish, PDELU, Swish, Soft Learnable, and more. For a detailed and comprehensive list of these activation functions, please refer to the full details available in Nanni et al. [57].

The main advantage of complex activation functions with learnable parameters is their ability to capture abstract features through non-linear transformations, a characteristic commonly observed in shallow networks [65]. However, a potential drawback lies in their complexity: multiple learnable parameters require large datasets for training.

3.2. Ensemble through Stochastic Approach

The stochastic approach [57] is used to alter the activation functions in ResNet50 or in MobileNetV2. 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 the models, 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 datasets, but this is not the case in the proposed ensemble method.

3.3. Data Augmentation

During the training process, sets of networks are trained using different data-augmentation techniques [19,66]. 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. The basic idea of DCT- and PCA-based approaches is similar: both methods allow us to project the image into a subspace and then return to the original space; by inserting noise into the backprojection, we can create new images. An example of the outcome of the third DCT-based approach is depicted in Figure 5.
- 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.

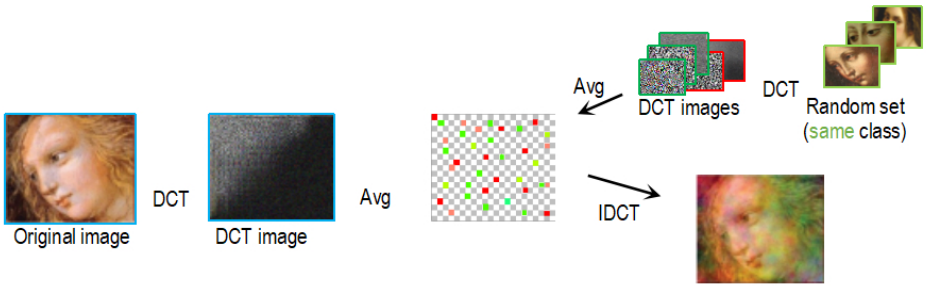

Figure 5. A sample image from APP5, where the image on the left represents the original image.

3.4. Parameter Ensembling via Perturbation (PEP)

Another approach for creating ensembles is PEP [67]. 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 [67], through experimentation on a validation set.

In practice, perturbations potentially allow a model to get out of local minima. In this way, we are able to create ensembles of different networks without having the computational cost of retraining a model.

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. This approach is described in Listing 1;
- 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. This approach is described in Listing 2;
- 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. This approach is described in Listing 3;
- PEPa: method similar to the original version, but where a small amount of random noise is injected. This approach is described in Listing 4;
- PEPb: the same idea as PEPa, but noise is injected in a different manner. This approach is described in Listing 5.

Essentially, the proposed methods apply the same idea as DCT-based data-augmentation methods by incorporating noise into the back projection. Different criteria are assumed to search for a new minimum in different areas of the solution space. DCTx and PEPx modify all parameters, while Dout either resets a parameter to zero or does not modify it. PEPx approaches perform noise injection on all parameters, DCTx on the other hand performs noise injection in the subspace and not in the original parameter space. 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 two epochs. 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.

Listing 1. Dout: it is 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 Dout.

```
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 Dout.

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

3.5. Wilcoxon Signed-Rank Test

The Wilcoxon signed-rank test, initially introduced by Mann and Whitney in 1947 [68], offers a statistical approach for comparing paired data samples derived from individual assessments. Unlike parametric tests, this non-parametric method does not hinge on assumptions concerning the underlying data distribution, such as normality. Instead, it considers both the magnitudes and directions of differences between paired observations.

Functioning as a non-parametric alternative to the paired student's *t*-test, the Wilcoxon signed-rank test proves especially valuable when population data does not conform to a normal distribution. Its primary objective lies in evaluating whether two related paired samples originate from identical distributions. By examining the ranks assigned to the disparities between paired observations, the Wilcoxon signed-rank test offers a robust approach for assessing the null hypothesis.

4. Experimental Results

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

4.1. Datasets

In our study, we utilized different datasets to evaluate the performance of our approach. Information for each dataset is presented in Table 1, including a brief identifier, the count of classes and samples, details on the testing protocol, and the original source reference. In the case of testing protocols, we utilize the corresponding abbreviation. In particular, *x*CV indicates that an *x*-fold cross-validation has been adopted (e.g., 10CV means that a 10-fold cross-validation has been used).

Table 1. Description of the datasets used in this study.

Short Name	# Classes	# Samples	Image Size	Protocol	Ref.
HE	10	862	grayscale	5CV	[69]
MA	4	257	grayscale	5CV	[70]
BG	3	300	RGB	5CV	[71]
LAR	4	1320	RGB	3CV	[72]
POR	3	1736	RGB	5CV	[73]
PEST	10	563	RGB	5CV	[74]
InfLAR	4	720	RGB	10CV	[75]
TRIZ	4	574	RGB	10CV	[76]

The following datasets have been adopted during the experiments:

- HE (2D HeLa dataset [69]): This contains fluorescence microscopy images of HeLa cells stained with different fluorescent dyes specific to various organelles. The dataset 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.
- MA (C. elegans Muscle Age dataset [70]): This dataset 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 [71]): This dataset, obtained from Zenodo (record: 834910#.Wp1bQ-jOWUl), has 300 annotated histological images of breast tissues from patients diagnosed with invasive ductal carcinoma. The dataset is categorized into three classes representing different grades (1–3) of carcinoma. A 5-fold cross-validation is applied.
- LAR (Laryngeal dataset [72]): Obtained from Zenodo (record: 1003200#.WdeQc-nBx0nQ), this has 1320 images of laryngeal tissues. It includes both healthy and early-stage cancerous tissues, representing a total of four tissue classes. This dataset is split into three folds by the original authors.
- The POR (portrait dataset) dataset [73] 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 (accessed on 23 October 2023).
- PEST [74] 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 [75] 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 [76] is a dataset of 574 gastric lesion-type images, four classes; as suggested by the original authors, we apply a 10-fold cross-validation.

By utilizing these diverse datasets, 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, we compare the performance of a single ResNet50 as well as different ensembles. In this set of experiments, we also checked whether the adoption of data augmentation affects the system's performance. In particular, Table 2 reports the results of this set of experiments. In this section, we use the following abbreviations:

- noDA: no data augmentation has been applied during training;
- DA: APP3 data augmentation has been applied during training;
- RE(x): a combination by sum rule of x standard ResNet50, where each network is simply re-trained on the training set;
- SE(x): a combination by sum rule of x networks coupled with the stochastic approach for replacing the activation function layers;
- StocDA_PEP(18): 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): for each DA method, we train three SE networks for a total of eighteen.

The first important result that can be inferred from Table 2 is that the baseline (i.e., a standard ReseNet50 with no data augmentation) always underperforms with respect to the ensembles. This is a piece of evidence about the fact that combining models into ensembles is beneficial. Another interesting observation is that the baseline benefits from data augmentation. This can be noticed by comparing the performance of RE(1)-noDA and RE(1)-DA. Data augmentation (at least the one adopted, i.e., APP3) improves the performance of the baseline, possibly introducing more information during the training phase.

Moreover, the results reported in Table 2 show that data augmentation is always useful for both RE and SE. Instead, for performance, the size of the ensemble seems to be important up to some extent as for both RE and SE the increase in the performance is poor when we increase the number of models from $x = 14$ to $x = 30$. We did not look for the optimal size of the ensemble as this may differ based on the dataset. The experiments

bring pieces of evidence that increasing the size of the ensemble at some point reaches a plateau where the gain in performance is poor compared with the number of added models. Moreover, ensembles built using SE always outperform ensembles that employ RE.

Table 2. Performance on the different datasets. The best accuracy (in %) for each dataset is in bold.

	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**

In Table 3, we compare the adoption of different data augmentation during the training phase. In this case, there is no clear winner. In each dataset, the rank position of the single data-augmentation method varies.

Table 3. Data-augmentation approaches. The best accuracy (in %) for each dataset is in bold.

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 4, 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 4. PEP variants. The best accuracy (in %) for each dataset is in bold.

	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 5 and 6, we compare different approaches for building an ensemble of ResNet50s. In particular, in Table 5, we report the accuracy of the different models and in Table 6 we report the error under the ROC curve (EUC) (this is the percentage defined as $(100 - (\text{Area under the ROC curve}))$.

Table 5. Accuracy (in %) of the different ensembles. The best accuracy for each dataset is in bold.

	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 6. EUC (in %) of the different ensembles. The best value for each dataset is in bold.

	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**

The comparison of the performance of the different ensembles allows us to highlight once again the fact that combining models into ensembles is beneficial, as the performance of the ensembles outperforms the single model. To better frame these results, we considered the well-known Wilcoxon-signed rank test to check whether these results are statistically significant for both the performance indicators (i.e., accuracy and EUC). The test proved that the results are significant as 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. Another fact that emerges from Tables 5 and 6 is that some ensembles are useful in datasets. For instance, when images are in grayscale, StocDA_PEP(18) and StocDA(18) are less performant. This is probably due to the fact that some data-augmentation methods are suitable for color images, whereas in LAR, POR, PEST, InfLAR, and TRIZ they fare better.

The limitations of ensembles are related to the reliance on available datasets and the significant computational resources required for their utilization. When an application necessitates heightened performance at the cost of computational power, the adoption of distillation techniques becomes imperative. Nevertheless, as evidenced by the data provided in Table 7, the execution times seem feasible for a wide range of applications.

Table 7. Inference time of a batch size of 100 images.

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

Now, to better validate our proposal, some further experiments are reported, see Tables 8–11:

- A different topology, mobilenetV2 [77], is used to assess the performance of the tested ensembles;
- To apply bagging for building the ensemble, we coupled bagging with RE(18)-DA, naming it Bag_RE(18)-DA.

To reduce the computation time, the following tests, Tables 8–11, were performed using only four datasets: HE; LAR; PEST; POR.

We reach the following conclusions:

- RE(18)-DA outperforms RE(1)-DA in all the datasets;
- SE(18)-DA outperforms RE(18)-DA in all the datasets but PEST;
- StocDA(18) outperforms SE(18)-DA in all the datasets but HE;
- StocDA_PEP(18) outperforms StocDA(18) in all the datasets but Pest when the accuracy is considered as performance indicator;
- Bagging does not lead to improvement; performance of RE(18)-DA and Bag_RE(18)-DA is similar.

Using only four datasets, we cannot do a reliable statistical analysis, but the trend is clear and it shows the usefulness of the ensemble.

Table 8. Comparison of accuracy (in %) of different ensembles with MobileNet. The best value for each dataset is in bold.

	HE	LAR	PEST	POR
RE(1)-DA	95.93	94.77	92.54	84.79
RE(18)-DA	95.93	95.91	93.26	86.72
SE(18)-DA	96.40	95.91	93.04	89.09
StocDA(18)	95.81	96.06	**93.31**	89.31
StocDA_PEP(18)	**96.51**	**96.29**	92.65	**90.17**

Table 9. Comparison of EUC (in %) of different ensembles with MobileNet. The best value for each dataset is in bold.

	HE	LAR	PEST	POR
RE(1)-DA	0.25	0.43	0.89	3.38
RE(18)-DA	**0.17**	0.25	0.67	2.57
SE(18)-DA	**0.17**	0.17	0.73	1.76
StocDA(18)	0.21	0.16	0.65	1.46
StocDA_PEP(18)	0.20	**0.13**	**0.62**	**1.23**

Table 10. Bagging-based approaches: accuracy (in %). The best value for each dataset is in bold.

	HE	LAR	PEST	POR
RE(18)-DA	96.33	**95.61**	**93.87**	**88.13**
Bag_RE(18)-DA	**96.40**	95.08	93.48	88.02

Table 11. Bagging-based approaches: EUC in (%). The best value for each dataset is in bold.

	HE	LAR	PEST	POR
RE(18)-DA	**0.22**	**0.18**	0.75	**2.05**
Bag_RE(18)-DA	0.31	0.19	**0.47**	2.30

5. Conclusions

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 to compare 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 a set of challenging image datasets encompassing diverse

tasks. As further topology, the mobilenetV2 was tested, confirming the conclusions obtained using ResNet50.

The experimental results demonstrated that the proposed ensemble of CNNs outperformed standalone methods for building CNN ensembles. As we compare the effects of various data-augmentation methods, it becomes clear that there is no noticeable pattern dictating the improvement in classification performance. Additional research is necessary to investigate the performance benefits of this approach. This could be pursued on a broader range of datasets (such as Computer Tomography (CT), Magnetic Resonance Imaging (MRI), and image/tumor segmentation), or adopting different network topologies.

In this last direction, as a future work, transformer topologies will also be tested, for instance, adopting Vit [78].

Conducting such investigations poses challenges, however, due to the substantial computational resources required for CNN and Transformer 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 A.L.; software, L.N.; writing—original draft preparation, S.B., A.L. and L.N.; writing—review and editing, S.B., A.L., L.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All the resources required to replicate our experiments are available at https://github.com/LorisNanni (accessed on 23 October 2023).

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. Wei, W.; Khan, A.; Huerta, E.; Huang, X.; Tian, M. Deep learning ensemble for real-time gravitational wave detection of spinning binary black hole mergers. *Phys. Lett. B* **2021**, *812*, 136029. [CrossRef]
2. Nanni, L.; Brahnam, S.; Lumini, A.; Loreggia, A. Coupling RetinaFace and Depth Information to Filter False Positives. *Appl. Sci.* **2023**, *13*, 2987. [CrossRef]
3. Shehab, M.; Abualigah, L.; Shambour, Q.; Abu-Hashem, M.A.; Shambour, M.K.Y.; Alsalibi, A.I.; Gandomi, A.H. Machine learning in medical applications: A review of state-of-the-art methods. *Comput. Biol. Med.* **2022**, *145*, 105458. [CrossRef]
4. Dutta, P.; Sathi, K.A.; Hossain, M.A.; Dewan, M.A.A. Conv-ViT: A Convolution and Vision Transformer-Based Hybrid Feature Extraction Method for Retinal Disease Detection. *J. Imaging* **2023**, *9*, 140. [CrossRef] [PubMed]
5. Wu, Z.; Tang, Y.; Hong, B.; Liang, B.; Liu, Y. Enhanced Precision in Dam Crack Width Measurement: Leveraging Advanced Lightweight Network Identification for Pixel-Level Accuracy. *Int. J. Intell. Syst.* **2023**, *2023*, 9940881. [CrossRef]
6. Deng, G.; Huang, T.; Lin, B.; Liu, H.; Yang, R.; Jing, W. Automatic meter reading from UAV inspection photos in the substation by combining YOLOv5s and DeepLabv3+. *Sensors* **2022**, *22*, 7090. [CrossRef] [PubMed]
7. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*. [CrossRef]
8. Haggenmüller, S.; Maron, R.C.; Hekler, A.; Utikal, J.S.; Barata, C.; Barnhill, R.L.; Beltraminelli, H.; Berking, C.; Betz-Stablein, B.; Blum, A.; et al. Skin cancer classification via convolutional neural networks: systematic review of studies involving human experts. *Eur. J. Cancer* **2021**, *156*, 202–216. [CrossRef]
9. 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. [CrossRef]
10. Horie, Y.; Yoshio, T.; Aoyama, K.; Yoshimizu, S.; Horiuchi, Y.; Ishiyama, A.; Hirasawa, T.; Tsuchida, T.; Ozawa, T.; Ishihara, S.; et al. Diagnostic outcomes of esophageal cancer by artificial intelligence using convolutional neural networks. *Gastrointest. Endosc.* **2019**, *89*, 25–32. [CrossRef]
11. 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. [CrossRef]
12. 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. *Front. Neurosci.* **2020**, *14*, 259. [CrossRef]

13. Nanni, L.; Loreggia, A.; Lumini, A.; Dorizza, A. A Standardized Approach for Skin Detection: Analysis of the Literature and Case Studies. *J. Imaging* **2023**, *9*, 35. [CrossRef]
14. Nagaraj, P.; Subhashini, S. A Review on Detection of Lung Cancer Using Ensemble of Classifiers with CNN. In Proceedings of the 2023 2nd International Conference on Edge Computing and Applications (ICECAA), Namakkal, India, 19–21 July 2023; pp. 815–820.
15. Shah, A.; Shah, M.; Pandya, A.; Sushra, R.; Sushra, R.; Mehta, M.; Patel, K.; Patel, K. A Comprehensive Study on Skin Cancer Detection using Artificial Neural Network (ANN) and Convolutional Neural Network (CNN). *Clin. eHealth* **2023**, *6*, 76–84. [CrossRef]
16. 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. In Proceedings of the 2020-5th International Conference on Information Technology (InCIT), Chonburi, Thailand, 21–22 October 2020; pp. 300–305.
17. 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. *Inf. Sci.* **2021**, *545*, 771–790. [CrossRef]
18. Panch, T.; Mattie, H.; Celi, L.A. The "inconvenient truth" about AI in healthcare. *NPJ Digit. Med.* **2019**, *2*, 77. [CrossRef]
19. Bravin, R.; Nanni, L.; Loreggia, A.; Brahnam, S.; Paci, M. Varied Image Data Augmentation Methods for Building Ensemble. *IEEE Access* **2023**, *11*, 8810–8823. [CrossRef]
20. Claro, M.L.; de MS Veras, R.; Santana, A.M.; Vogado, L.H.S.; Junior, G.B.; de Medeiros, F.N.; Tavares, J.M.R. Assessing the Impact of Data Augmentation and a Combination of CNNs on Leukemia Classification. *Inf. Sci.* **2022**, *609*, 1010–1029. [CrossRef]
21. Nanni, L.; Fantozzi, C.; Loreggia, A.; Lumini, A. Ensembles of Convolutional Neural Networks and Transformers for Polyp Segmentation. *Sensors* **2023**, *23*, 4688. [CrossRef]
22. 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: Amsterdam, The Netherlands, 2023; pp. 215–234.
23. 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. [CrossRef]
24. Yao, X.; Liu, Y. Making use of population information in evolutionary artificial neural networks. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **1998**, *28*, 417–425.
25. Opitz, D.; Shavlik, J. Generating accurate and diverse members of a neural-network ensemble. *Adv. Neural Inf. Process. Syst.* **1995**, *8*.
26. Liu, Y.; Yao, X.; Higuchi, T. Evolutionary ensembles with negative correlation learning. *IEEE Trans. Evol. Comput.* **2000**, *4*, 380–387.
27. Rosen, B.E. Ensemble learning using decorrelated neural networks. *Connect. Sci.* **1996**, *8*, 373–384. [CrossRef]
28. Liu, Y.; Yao, X. Ensemble learning via negative correlation. *Neural Netw.* **1999**, *12*, 1399–1404. [CrossRef] [PubMed]
29. 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. In Proceedings of the Medical Imaging 2020: Computer-Aided Diagnosis, Houston, TX, USA, 16–19 February 2020; Volume 11314, pp. 228–235.
30. He, X.; Zhou, Y.; Wang, B.; Cui, S.; Shao, L. Dme-net: Diabetic macular edema grading by auxiliary task learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Cham, Switzerland, 2019; pp. 788–796.
31. 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. [CrossRef] [PubMed]
32. Savelli, B.; Bria, A.; Molinara, M.; Marrocco, C.; Tortorella, F. A multi-context CNN ensemble for small lesion detection. *Artif. Intell. Med.* **2020**, *103*, 101749. [CrossRef]
33. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778. [CrossRef]
34. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
35. Matloob, F.; Ghazal, T.M.; Taleb, N.; Aftab, S.; Ahmad, M.; Khan, M.A.; Abbas, S.; Soomro, T.R. Software defect prediction using ensemble learning: A systematic literature review. *IEEE Access* **2021**, *9*, 98754–98771. [CrossRef]
36. Roshan, S.E.; Asadi, S. Improvement of Bagging performance for classification of imbalanced datasets using evolutionary multi-objective optimization. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103319. [CrossRef]
37. 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* **2019**, arXiv:1909.11870.
38. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
39. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
40. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2818–2826.
41. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.

42. Liu, K.; Zhang, M.; Pan, Z. Facial expression recognition with CNN ensemble. In Proceedings of the 2016 International Conference on Cyberworlds (CW), Chongqing, China, 28–30 September 2016; pp. 163–166.
43. Goodfellow, I.J.; Erhan, D.; Carrier, P.L.; Courville, A.; Mirza, M.; Hamner, B.; Cukierski, W.; Tang, Y.; Thaler, D.; Lee, D.H.; et al. Challenges in representation learning: A report on three machine learning contests. In Proceedings of the Neural Information Processing: 20th International Conference, ICONIP 2013, Daegu, Republic of Korea, 3–4 November 2013; Proceedings, Part III 20; Springer: Cham, Switzerland, 2013; pp. 117–124.
44. Kumar, A.; Kim, J.; Lyndon, D.; Fulham, M.; Feng, D. An ensemble of fine-tuned convolutional neural networks for medical image classification. *IEEE J. Biomed. Health Inform.* **2016**, *21*, 31–40. [CrossRef]
45. 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 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
46. Gilbert, A.; Piras, L.; Wang, J.; Yan, F.; Ramisa, A.; Dellandrea, E.; Gaizauskas, R.J.; Villegas, M.; Mikolajczyk, K. Overview of the ImageCLEF 2016 Scalable Concept Image Annotation Task. In Proceedings of the CLEF (Working Notes), Évora, Portugal, 5–8 September 2016; pp. 254–278.
47. Pandey, P.; Deepthi, A.; Mandal, B.; Puhan, N.B. FoodNet: Recognizing foods using ensemble of deep networks. *IEEE Signal Process. Lett.* **2017**, *24*, 1758–1762. [CrossRef]
48. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
49. Wolpert, D.H.; Macready, W.G. An efficient method to estimate bagging's generalization error. *Mach. Learn.* **1999**, *35*, 41–55. [CrossRef]
50. Bauer, E.; Kohavi, R. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Mach. Learn.* **1999**, *36*, 105–139. [CrossRef]
51. Kim, P.K.; Lim, K.T. Vehicle type classification using bagging and convolutional neural network on multi view surveillance image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 41–46.
52. Dong, X.; Qian, L.; Huang, L. A CNN-based bagging learning approach to short-term load forecasting in smart grid. In Proceedings of the 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), San Francisco, CA, USA, 4–8 August 2017; pp. 1–6.
53. Guo, J.; Gould, S. Deep CNN ensemble with data augmentation for object detection. *arXiv* **2015**, arXiv:1506.07224.
54. Gan, Y.; Chen, J.; Xu, L. Facial expression recognition boosted by soft label with a diverse ensemble. *Pattern Recognit. Lett.* **2019**, *125*, 105–112. [CrossRef]
55. Antipov, G.; Berrani, S.A.; Dugelay, J.L. Minimalistic CNN-based ensemble model for gender prediction from face images. *Pattern Recognit. Lett.* **2016**, *70*, 59–65. [CrossRef]
56. Zhang, H.; Zhou, T.; Xu, T.; Hu, H. Remote interference discrimination testbed employing AI ensemble algorithms for 6G TDD networks. *Sensors* **2023**, *23*, 2264. [CrossRef]
57. Nanni, L.; Lumini, A.; Ghidoni, S.; Maguolo, G. Stochastic selection of activation layers for convolutional neural networks. *Sensors* **2020**, *20*, 1626. [CrossRef] [PubMed]
58. Ju, C.; Bibaut, A.; van der Laan, M. The relative performance of ensemble methods with deep convolutional neural networks for image classification. *J. Appl. Stat.* **2018**, *45*, 2800–2818. [CrossRef]
59. Harangi, B. Skin lesion classification with ensembles of deep convolutional neural networks. *J. Biomed. Inform.* **2018**, *86*, 25–32. [CrossRef] [PubMed]
60. Lyksborg, M.; Puonti, O.; Agn, M.; Larsen, R. An ensemble of 2D convolutional neural networks for tumor segmentation. In Proceedings of the Image Analysis: 19th Scandinavian Conference, SCIA 2015, Copenhagen, Denmark, 15–17 June 2015; Proceedings 19; Springer: Cham, Switzerland, 2015; pp. 201–211.
61. Minetto, R.; Segundo, M.P.; Sarkar, S. Hydra: An ensemble of convolutional neural networks for geospatial land classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6530–6541. [CrossRef]
62. Dong, X.; Yu, Z.; Cao, W.; Shi, Y.; Ma, Q. A survey on ensemble learning. *Front. Comput. Sci.* **2020**, *14*, 241–258.
63. Brown, G.; Wyatt, J.; Harris, R.; Yao, X. Diversity creation methods: A survey and categorisation. *Inf. Fusion* **2005**, *6*, 5–20. [CrossRef]
64. Sagi, O.; Rokach, L. Ensemble learning: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1249. [CrossRef]
65. Duch, W.; Jankowski, N. Survey of neural transfer functions. *Neural Comput. Surv.* **1999**, *2*, 163–212.
66. Goceri, E. Medical Image Data Augmentation: Techniques, Comparisons and Interpretations. *Artif. Intell. Rev.* **2023**, *7*, 1–45.
67. Mehrtash, A.; Abolmaesumi, P.; Golland, P.; Kapur, T.; Wassermann, D.; Wells, W. Pep: Parameter ensembling by perturbation. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 8895–8906. [PubMed]
68. Demšar, J. Statistical Comparisons of Classifiers over Multiple Datasets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30. [CrossRef]
69. 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. [PubMed]
70. 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.

71. 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.
72. 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*, 034502.
73. Kim, Y.W.; Byun, Y.C.; Krishna, A.V.N. Portrait Segmentation Using Ensemble of Heterogeneous Deep-Learning Models. *Entropy* **2021**, *23*, 197. [CrossRef]
74. Deng, L.; Wang, Y.; Han, Z.; Yu, R. Research on insect pest image detection and recognition based on bio-inspired methods. *Biosyst. Eng.* **2018**, *169*, 139–148.
75. 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. *Med. Biol. Eng. Comput.* **2020**, *58*, 1225–1238.
76. Zhao, R.; Zhang, R.; Tang, T.; Feng, X.; Li, J.; Liu, Y.; Zhu, R.; Wang, G.; Li, K.; Zhou, W.; et al. TriZ-a rotation-tolerant image feature and its application in endoscope-based disease diagnosis. *Comput. Biol. Med.* **2018**, *99*, 182–190. [CrossRef]
77. Sandler, M.; Howard, A.G.; Zhu, M.; Zhmoginov, A.; Chen, L. Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation. *arXiv* **2018**, arXiv:1801.04381.
78. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.