Soft Computing manuscript No.
(will be inserted by the editor)

Ensemble classification from deep predictions with test data augmentation

Jorge Calvo-Zaragoza · Juan R. Rico-Juan · Antonio-Javier Gallego

Received: date / Accepted: date

Abstract Data augmentation has become a standard step to improve the predictive power and robustness of Convolutional Neural Networks by means of the synthetic generation of new samples depicting different deformations. This step has been traditionally considered to improve the network at the training stage. In this work, however, we study the use of data augmentation at classification time. That is, the test sample is augmented, following the same procedure considered for training, and the decision is taken with an ensemble prediction over all these samples. We present comprehensive experimentation with several datasets and ensemble decisions, considering a rather generic data augmentation procedure. Our results show that performing this step is able to boost the original classification, even when the room for improvement is limited.

Keywords Convolutional Neural Networks · Data augmentation · Ensemble classification

First author thanks the support from the Spanish Ministerio de Ciencia, Innovación y Universidades through Juan de la Cierva - Formación grant (Ref. FJCI-2016-27873).

Jorge Calvo-Zaragoza
Department of Software and Computing Systems, University of Alicante
Carretera San Vicente del Raspeig s/n, 03690 Alicante, Spain
Tel.: +349-65-903772
E-mail: jcalvo@dlsi.ua.es

Juan R. Rico-Juan
Department of Software and Computing Systems, University of Alicante
Carretera San Vicente del Raspeig s/n, 03690 Alicante, Spain

Antonio-Javier Gallego
Department of Software and Computing Systems, University of Alicante
Carretera San Vicente del Raspeig s/n, 03690 Alicante, Spain

1 Introduction

Deep convolutional neural networks have been one of the biggest breakthroughs in the field of Pattern Recognition and Image Analysis [23]. These networks allow learning a hierarchy of features suitable for the recognition task by means of a series of stacked convolutional layers. Although these networks were initially proposed decades ago [24], several factors have contributed to their eventual success.

On the one hand, some of these contributions are of a technological nature: more powerful computational capabilities or an efficient use of graphical units; or of a logistical nature, like having bigger amounts of (labeled) data. Others, however, are purely methodological. For example, the use of activation functions that palliate the so-called vanishing gradient problem, such as the Rectified Linear Units [11], smart initialization of the network weights [32], modifications to the gradient descent scheme [9], and so on. Note, however, that most of these improvements have focused especially on solving technical issues, like those that make the network converge, or do it more efficiently.

On the other hand, another set of contributions have focused on making these networks be more accurate, or better said, generalize better and prevent over-fitting. An example of such contributions is *dropout*, a scheme that randomly disconnect neurons from the network at each step of the gradient descent [30]. Even some derivatives of this technique has been proposed — such as DropConnect [34] — and it is expected that this type of modifications continue to arise in the short-term.

Despite all of the above, the start of the Deep Learning era is often located at the work of [20], which presented the widely known *AlexNet*. This model, in addition to proposing a really deep convolutional network

(at least for that time, we can now find much deeper networks), showed the importance of what has been called *data augmentation*.

Data augmentation is a step focused on generating a set of synthetic samples out of those in the training set, with the aim of boosting the performance. Actually, data augmentation is related to the term perturbation-based methods, which also modify the input data. In some works, both augmentation and perturbation might be equivalent [13,26]. However, perturbation methods traditionally focus on variations of input features [1, 37,3], whereas data augmentation is considered at the whole sample level [12].

There are several ways to do data augmentation in images (rotation, color variation, random occlusions, etc.), although the goodness of each one is strongly dependent on the task at issue. The intention of this process is twofold: i) since these neural networks need to be trained on a large set of data, data augmentation might boost the performance by increasing the size of the original training set, ii) if the augmentation procedure creates examples that mimic expected distortions, the network might be more robust to variations at test stage. In addition, some proposals try to learn this augmentation from data to optimize the eventual performance [25].

The contribution of this work is to further study data augmentation, not only for training but also for the test stage. The main idea is that test samples are also augmented, producing a series of new synthetic test samples. The model has then to predict the category of the original input as well as the category of the augmented ones. The final decision is eventually taken considering all these predictions by means of some kind of ensemble mechanism. To demonstrate the goodness of this methodology, we perform exhaustive experiments with a number of neural configurations, image datasets, and ensemble strategies. The results, validated with statistical significance tests, show that this procedure allows a general improvement of the classification accuracy, even in those cases where there is little room for improvement.

The rest of the work is structured as follows: related works concerning ensemble decision and convolutional neural networks are described in Section 2; the ensemble prediction scheme using data augmentation is detailed in Section 3; the experimental setup is described in Section 4, while results are reported in Section 5. Finally, main conclusions are drawn in Section 6, along with some discussion about future work.

2 Related work

Classification systems have been widely studied within the pattern recognition field. The classical scheme is based on a sequential model that consists in extracting features from a sample, using a classification technique and obtaining a hypothesis [10]. This scheme has been exploited in order to attain fairly complex techniques with which to improve classification accuracy, such as Artificial Neural Networks [16] or Support Vector Machines [5]. The evolution in this field, however, has led to the development of new schemes in supervised learning.

For example, with the recent advances in deep neural networks, the task of extracting features is no longer a separate process that depends, in most cases, on users' expertise, but the network itself learns which features are most useful for the task at hand. From another point of view, several classification scheme emerged based on the assumption that it is more robust to combine a set of hypotheses than to use just one [17]. These schemes combine the scores of individual classifiers (usually called weak classifiers) to produce a final score. They are commonly referred to as ensemble classifiers. A wide analysis of this kind of algorithms can be found in the book of [21]. Recently, both scenarios have been considered together.

There are many ways to combine decisions under the convolutional neural network paradigm. The closest to what has traditionally been done is to train different neural networks whose predictions are combined at the time of classification. The differences among networks can be found at several levels: different topology — or same topology with different initialization —, different optimization algorithm or different training data. It can also be done through the network itself by establishing different convolution branches that are eventually joined by means of concatenation, summation or maxout [29].

In our case, we consider the well-establish data augmentation step to improve the capabilities of the network. Data augmentation has been systematically applied to improve the training of neural networks and make them more robust against distortions of the input. Since the popularity given by the work of Krizhevsky et al. [20] for improving object recognition, data augmentation has been applied in a wide variety of applications like acoustic modeling [7], speech recognition [18] or biometrics [31,2].

Our object of study, nonetheless, does not conflict with any of the mechanisms considered above, but rather serves as a complement to all of them. We consider the use of data augmentation at prediction time. For

each sample to be classified, a set of augmented samples is obtained. They are then processed by the network, yielding a series of predictions. Eventually, these predictions are combined, as would be done in any ensemble system.

Test data augmentation has been slightly considered in some previous works, focused on very specific tasks [27,28,14]. Our objective is, therefore, to study test data augmentation in a comprehensive way, so that general conclusions about this procedure can be drawn. Unlike previous works, we provide a proper formulation of the methodology, as well as a thorough experimentation to validate the approach with a number of datasets, different neural models, many data augmentation strategies, and several ways of combining the neural decisions. In addition, our results are validated with statistical significance tests in order to minimize the possibility that the differences in accuracies are due to chance variation.

3 Ensemble of deep predictions with data augmentation

The scheme studied in this work is very simple to apply, which, in addition to the good results it provides, makes it a very good tool to improve the predictive power of Convolutional Neural Networks. To apply it to a particular task, the recipe involves three elements: a labeled training set, a network topology, and a data augmentation procedure that allows generating an arbitrary number of new examples. Note that this is indeed a very common scenario nowadays.

First, the network is trained in a supervised way using the training set. In spite of not being strictly required, the training stage can be carried out applying data augmentation to the training set. Then, at the time of classifying an input query, the process entails three steps:

1. The sample is augmented — with the same procedure considered previously — until obtaining T new elements.
2. Each of the $T+1$ samples is predicted by the trained network.
3. The decision about the actual sample received is taken with all the predictions made, following some kind of combination strategy.

A summary of this process is illustrated in Fig. 1.

The last step needs a strategy to combine the different predictions computed by the network. Let Ω denote the categories of the task. Let x_0 be the test query, and $x_1 \dots x_T$ represent the T generated test samples from x_0. Let $P(w|x_i)$ denote the probability that the network gives x_i to belong to class w. If we denote $\hat{w}$ as the decision finally taken by the ensemble, we consider in this work the following policies:

- Average (avg). The label predicted is that which maximizes the average probability among the predictions for each sample.

$$\hat{w} := \arg\max_{w \in \Omega} \frac{\sum_{i=0}^{T} P(w|x_i)}{T+1}$$

- Maximum (max). It proposes the label for which the maximum probability is reached in any of the predictions.

$$\hat{w} := \arg\max_{w \in \Omega} \max_{0 \le i \le T} P(w|x_i)$$

- Mode (mode). It takes the class which receives the majority of votes among the different predictions.

$$\hat{w} := \arg\max_{w \in \Omega} \sum_{i=0}^{T} \mathbf{1}_{w = \arg\max_{w' \in \Omega} P(w'|x_i)}$$

Our premise is that it might exist some samples that are noisy — that is, they can be easily confused with other categories — yet if we apply data augmentation in the same way as done during training, it is more unlikely that all the augmented samples also look similar to the same different category. Thus, taking all these samples into account, a more robust prediction must be obtained.

Below we show a series of experiments in which it is verified that this procedure somehow allows improving the classification figures.

4 Experimental setup

The present work aims at performing an analysis that can be generalizable to a wide range of applications but, specifically, those whose entries can be represented by images. For this reason, we consider several datasets of this nature, a rather general data augmentation procedure and a series of network topologies. A description of the experimental setup followed is presented in this section.

4.1 Datasets

Although deep neural networks can be used in a wide variety of problems, we restrict ourselves to considering image datasets. This decision has been taken for two reasons: on the one hand, data augmentation is more established for this type of tasks; on the other hand, this

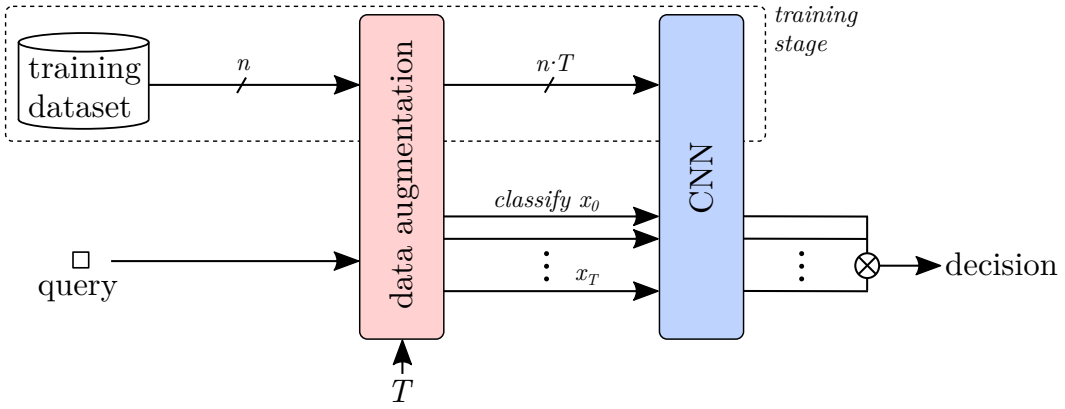


Fig. 1 Classification scheme studied in this work. The Convolutional Neural Network (CNN) is trained conventionally considering a data augmentation procedure. At the time of classification, this process is applied to the query and a prediction is made on all samples obtained. The final decision is taken with all the predictions following some type of aggregation.

Name	Instances	Classes	Shape
USPS	9 298	10	16×16
MNIST	10 000	10	28×28
MPEG-7	1 400	70	35×35
HOMUS	15 200	32	40×40
NIST	9 984	26	32×32
CIFAR10	10 000	10	32×32

Table 1 Description of the image datasets used in the experimentation.

kind of datasets allows us to use a general data augmentation procedure that can serve as a generic scenario.

A total of 6 datasets are considered: the *United States Postal Office* (USPS) [15] and *MNIST* [24] datasets of handwritten digits with binary images; the *MPEG-7* shape silhouette dataset [22]; the *Handwritten Online Musical Symbol* (HOMUS) [6], which depicts binary images of isolated handwritten music symbols; the *NIST SPECIAL DATABASE 19* (NIST) of isolated characters [35], from which a subset of the upper case characters was selected; and the reference dataset within the computer vision community *CIFAR-10* [19], consisting of 32×32 color images representing 10 different categories (subset of the *80 million tiny images* dataset [33]).

The details of these datasets are summarized in Table 1. In some specific cases (such as MNIST, NIST and CIFAR10) the number of samples used is lower than those available. The reason is to reduce the overload when running experiments with different values of data augmentation.

At each experiment carried out, a 5-fold cross-validation process has been applied to provide more robust figures with respect to the variance of the training data.

4.2 Data augmentation

The idea of our work is to present a scenario that can be generalizable to other situations. That is why we do not exhaustively search the best data augmentation for each considered dataset, which would obviously vary the figures. Instead, we consider a rather generic data augmentation process. Specifically, we make use of a simple procedure in which the possible types of deformations are the following ones:

- Rotation in the range of $[-30°, +30°]$. More aggressive rotations might conflict with characters.
- Horizontal and vertical translation in a range of $[-10\ \%, +10\ \%]$ with respect to each dimension of the image. Note that two deformations are actually performed in this case (one per each dimension).
- Zoom of $[-10\ \%, +10\ \%]$. When filling the gaps created by a negative zooming, the value of the closest pixel in the original image is taken.

This process takes an input image and produces T new images in which these transformations are applied at the same time, choosing values at random within the range of each type. Note that all ranges considered admit a zero value that means that no transformation is applied. An example of augmented data is illustrated in Fig. 2.

4.3 Network models

In order to perform a comprehensive analysis of the results, we use three different network topologies based on its size and depth. A brief description of each one is presented below:

- Small: it consist of two stacked convolutional layers with 3×3 kernel and 32 filters, followed by a $2 \times$

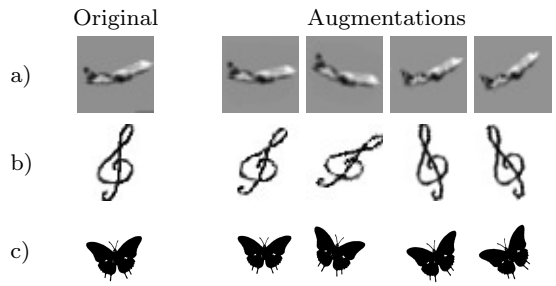


Fig. 2 Examples of data augmentation applied to data CIFAR (a), HOMUS (b), and MPEG-7 (c) datasets. The generated samples might depict several deformations at the same time, namely rotation, zoom and translation.

2 max-pooling and dropout of 25%. Then, a fully-convolutional layer with 128 units and dropout of 50% is added.
- Medium: it begins with two blocks consisting of two convolutional plus 2×2 max-pooling plus dropout (25%) layers, with 64 filters and 3×3 kernel and 128 filters and 5×5 kernel, respectively. These blocks are followed by two fully-connected layers of 32 units, with dropout of 50 %.
- Advanced: it comprises two blocks of 3×3 convolution plus 2×2 max-pooling plus 20 % of dropout, with 256 and 128 filters, respectively. Then, two blocks of 3×3 convolution plus 20 % of dropout, with 128 and 64 filters, respectively. The final block comprises three fully-connected layers with 10 % of dropout, with 512, 256 and 128 units, respectively.

These network have been fixed after slightly tuning their configurations for improving the accuracy in the considered corpora.

The learning of the network weights is performed by means of stochastic gradient descent [4], considering the adaptive learning rate proposed by Zeiler [36].

5 Results

In this section we present the results obtained in our experiments. From now on, T_{train} denotes the factor of data augmentation at training (the number of samples obtained from an original one with the augmentation procedure). Analogously, T_{test} denotes the same factor during test.

The development for the first experiment is as follows: for each dataset, T_{train} factors of 0 (no augmentation) and 5 are applied with the generic procedure discussed in the previous section. For each case, we measure the classification error achieved by the three network models with test data augmentation factor ranging from 0 to 15, in which every ensemble decision scheme is computed.

The results of this experiment are shown in Table 2. For the sake of comparison, the best value obtained among the test data augmentation factors is shown in each case. Column *orig* means the original classification, *ie.* no data augmentation at test is performed.

An initial remark to begin with is that, as expected, the traditional use of data augmentation during training helps to boost the performance of the neural networks. However, data augmentation at the test stage does not succeed if it has not been considered data augmentation during training. This is seen in the fact that in most rows with $T_{\text{train}} = 0$, the best value obtained is in the *orig* columns. Furthermore, when data augmentation is considered during the training of the networks, the best results are then found when performing test data augmentation and combination during the prediction. Since these two procedures — data augmentation during both training and prediction — are not exclusive, but can be used simultaneously, the goodness of our proposal is clearly validated.

Concerning the specific procedure of test data augmentation, the *avg* method has been reported as the best way to combine decisions, which basically performs an average of the probability given to each class. In some cases, *mode* also manages to improve the figures (HOMUS, CIFAR10 or MPEG-7 with the advanced model) yet to a lesser extent. In contrast, *max* combination seems less adequate and only makes a significant contribution in one case (USPS with the advanced model).

As regards the network topology, the *small* model tends to not improve the original results, regardless of the training augmentation factor and the combination method, whereas the results do improve with test data augmentation and combination with *medium* and *advanced* models.

As a general conclusion from these experiments, the data augmentation at the classification stage is beneficial since the best results for each dataset (considering all cases) are obtained when this process is used. It does not overlap with the traditional procedure of data augmentation during training, but they are complementary.

In many cases, however, the improvement from test data augmentation is not numerically significant. The problem is that the conventional strategies already achieve good results in many of the considered datasets. That is why, in order to verify the goodness of the process, we carry out a more in-depth analysis on those datasets in which there is room for improvement, namely CIFAR10 and MPEG7.

		small				medium				advanced			
Dataset	T_{train}	**orig**	**avg**	**max**	**mode**	**orig**	**avg**	**max**	**mode**	**orig**	**avg**	**max**	**mode**
CIFAR10	**0**	45.63	**44.77**	47.52	45.30	**39.74**	40.44	41.57	41.14	**37.94**	38.32	39.60	39.14
	5	39.33	**36.83**	38.41	37.70	33.59	**30.38**	32.70	31.09	33.17	**31.14**	32.17	31.57
HOMUS	**0**	**12.86**	17.40	17.56	20.86	**8.09**	10.60	10.57	12.25	**8.20**	10.35	10.45	11.30
	5	8.40	**8.05**	8.72	8.27	4.73	**4.34**	4.84	4.39	4.47	**4.26**	4.49	4.27
MNIST	**0**	**2.20**	3.54	3.52	4.06	**1.61**	2.40	2.53	2.59	**1.54**	2.10	2.25	2.38
	5	**1.20**	1.42	1.50	1.49	**0.87**	**0.87**	1.00	**0.87**	0.71	**0.70**	0.73	0.73
MPEG7	**0**	**17.07**	18.57	18.50	42.86	**20.21**	22.07	22.50	39.29	**15.64**	20.00	20.00	36.64
	5	**13.00**	14.22	13.93	16.00	11.07	**11.00**	11.50	11.57	9.86	**8.93**	10.00	9.36
NIST	**0**	**5.96**	9.52	9.48	15.47	**3.83**	5.52	5.52	6.98	**3.73**	5.64	5.61	7.63
	5	**3.56**	3.98	4.21	4.31	2.40	**2.30**	2.47	2.35	2.38	**2.29**	2.55	2.40
USPS	**0**	**1.83**	2.98	2.92	8.14	**1.45**	2.18	2.17	5.70	**1.71**	2.83	2.86	4.34
	5	**1.26**	1.38	1.33	1.45	**0.91**	0.94	0.99	0.97	0.90	0.86	**0.82**	0.86
Average	**0**	**13.79**	16.21	16.27	22.30	**11.60**	13.25	13.29	17.00	**10.69**	12.65	12.68	15.82
	5	**10.62**	10.64	11.07	11.07	8.15	**7.67**	8.26	7.84	7.83	**7.39**	7.84	7.51

Table 2 Mean error rate over a 5-fold cross validation for each dataset considered (average results are also presented) and data augmentation during training with respect to the network model and way of combining decisions. Column *orig* means the original classification, *ie.* no data augmentation at test is performed. The figures presented depict the best lowest error obtained with test data augmentation ranging from 0 to 15.

First, since it is a differential factor, we increase the factor of data augmentation in the training set, considering values of 0, 5, 10 and 15. Table 3 presents the results of this new experiment. For the sake of readability, we restrict ourselves to the use of the *avg* ensemble combination, which has been reported as the best strategy for this scheme. It can be appreciated that the considered data augmentation procedure is indeed profitable but higher factors are not expected to decrease the error (see Fig. 3).

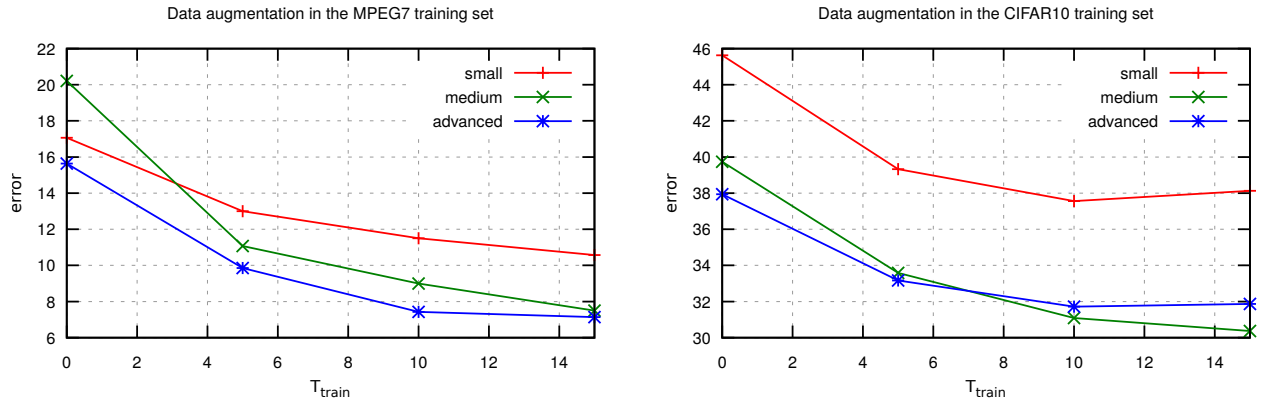
The first conclusion that can be drawn from this new experiment is that the augmentation at the training stage directly influences the goodness of the augmentation at the test stage. That is why there is only one case in which it is possible to improve with test data augmentation without doing augmentation during training (CIFAR10, small model). Intuitively, this phenomenon may be explained by a higher likelihood of augmented test data being more similar to some of the generated data during training.

It can be seen that the depth of the network model influences the figures but to a lesser extent. Amongst the cases including training data augmentation, only for MPEG7 with the small model the best prediction is just to classify the original samples. In the rest of the cases, a decreasing error tendency is seen as more samples in the test stage are combined. While the very first cases are not robust, giving irregular results, the best overall figures usually come from $T_{\text{test}} \geq 8$. The trend of these results is illustrated in Fig. 4 (CIFAR10) and 5 (MPEG-7). Note that to better appreciate the curves, the scales on the Y axis have been adjusted for each particular case.

Eventually, test data augmentation allows improving the performance of the classification for each dataset: in CIFAR10, from an error of 30.37 ($T_{\text{train}} = 15$, medium model) to 27.70 (same row, $T_{\text{test}} = 14$); in MPEG7, from an error of 7.24 ($T_{\text{train}} = 15$, advanced model) to 6.14 (same row, $T_{\text{test}} = 14$).

Although these improvements are not huge, it should be noted that they are achieved using exactly the same elements as those used in the conventional case: a Convolutional Neural Network, a training set, and an data augmentation process. It is, therefore, advantageous to consider the test data augmentation procedure to fully exploit the powerful of the classification scheme.

In addition, it is important to emphasize that the specific procedure of data augmentation has proven to play a fundamental role in this process. In our work, we have used a generic data augmentation to serve uniformly for all datasets considered. The fact of considering task-dependent data augmentation is likely to improve the figures attained in this work.



(a) MPEG-7 dataset

(b) CIFAR10 dataset

Fig. 3 Impact of data augmentation during training for MPEG7 and CIFAR10 datasets.

5.1 Statistical significance tests

Previous section presented the average figures obtained in our experiments. It allowed us to analyze the general trend of the study. However, statistical tests are considered in this section for comparing the results objectively [8]. Specifically, Wilcoxon signed-rank tests are used, which allow comparing results obtained by pairs. To perform these tests, the results obtained in each folder of the considered datasets are used, amalgamating the network models.

Table 4 reports the results of such test for the first experiment, with a significance established to 0.05. The figures considered for computing the test are those that represent the best factor of test data augmentation for each case. Symbol ✓ state that the test is significantly accepted, that is, that the results obtained by the scheme in the row is significantly better than the ones obtained by the scheme in the column.

These statistical results clarify the aforementioned trends. When there is no data augmentation in training, the data augmentation in prediction is detrimental (top of the table). Furthermore, it can also be seen that the augmentation in training effectively improves the results obtained in any condition (bottom-left of the table). The most interesting part is that in which the schemes are compared with augmentation in both training and prediction (bottom-right): in this case, it is reflected that the test data augmentation with *avg* combination allows a significant improvement with respect to the other schemes. The combination *mode* also improves the original results, whereas the combination *max* does not imply any significant improvement.

In the second experiment we focused on increasing the factor of training data augmentation to measure its impact on the results with a test data augmentation factor ranging from 1 to 15. Recalling from the previous section, only *avg* combination was considered because of its superior performance against other combinations. Table 5 reports the statistical tests restricted to the best value of test data augmentation for each case.

It can be observed that as the factor of training data augmentation is increased, results are significantly better than their peers. In these cases, data augmentation at prediction time is also always beneficial. It is interesting to observe that even in some cases, test data augmentation significantly outperforms the results against a higher factor of training data augmentation, as happens with $T_{\text{train}} = 10$ *avg* against $T_{\text{train}} = 15$ *orig*).

6 Conclusions

This paper studies the use of data augmentation at the time of classification. This strategy consists in considering the generation of augmented data from a given test query, whose classification is made with a combination of the predictions obtained with all the samples (the original and the generated ones). We present comprehensive experimentation with several types of combinations, image datasets, convolutional network models, and general data augmentation procedures.

Our results show a direct relationship between the training and test data augmentation. Without the first, the second seems to be harmful. However, if the same data augmentation is considered in both cases, the classification can be improved with respect to the conventional scenario. These results have been validated through statistical significance tests, thereby demon-

			T_{test}														
Dataset	T_{train}	**model**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
CIFAR10	**0**	**small**	45.63	47.68	47.01	46.21	45.80	45.84	45.41	45.06	45.08	44.95	44.90	44.79	44.95	44.94	**44.77**	44.82
		medium	**39.74**	41.58	42.45	41.93	41.77	41.54	41.11	41.01	41.03	40.92	40.80	40.74	40.48	40.60	40.48	40.44
		advanced	**37.94**	39.52	39.62	39.50	39.60	39.35	39.15	39.00	38.93	38.90	38.49	38.32	38.51	38.52	38.35	38.41
	5	**small**	39.33	38.99	39.24	38.63	38.68	38.15	38.24	37.76	37.59	37.29	37.26	37.20	36.98	**36.83**	36.98	36.96
		medium	33.59	33.16	33.06	32.21	32.04	31.82	31.61	31.50	31.24	31.40	31.09	31.03	30.67	30.51	**30.38**	30.59
		advanced	33.17	33.36	32.89	32.22	32.28	31.97	31.62	31.71	31.34	31.59	31.45	31.31	31.30	31.32	31.27	**31.14**
	10	**small**	37.56	37.76	37.53	37.22	37.05	36.92	36.79	36.57	36.21	36.49	36.42	36.47	36.35	36.38	36.20	**36.17**
		medium	31.09	30.57	30.51	29.89	29.08	29.05	28.55	28.66	28.65	28.40	**28.23**	28.47	**28.23**	28.30	28.31	28.37
		advanced	31.72	31.71	31.41	31.11	30.63	30.55	29.96	29.88	29.80	29.80	29.69	29.67	29.70	29.59	29.53	**29.43**
	15	**small**	38.13	37.70	37.71	37.47	36.89	36.65	36.71	36.67	**36.51**	36.60	36.61	36.57	36.67	36.61	36.67	36.77
		medium	30.37	29.85	29.56	29.12	29.01	28.76	28.35	28.06	28.10	28.07	27.87	28.08	27.91	27.88	**27.70**	27.73
		advanced	31.87	31.35	30.98	30.64	30.36	30.01	29.96	30.30	29.98	30.01	29.86	29.89	29.83	29.75	29.64	**29.60**
MPEG7	**0**	**small**	**17.07**	18.57	24.93	30.93	34.79	37.64	39.28	40.29	41.07	40.79	41.57	42.07	41.64	41.71	42.07	41.86
		medium	**20.21**	22.07	24.93	26.71	28.07	29.72	30.57	30.93	31.43	31.78	32.14	32.93	33.14	32.79	33.00	32.21
		advanced	**15.64**	20.00	28.29	32.14	33.00	33.14	33.43	33.21	34.00	34.50	34.64	33.79	34.14	34.36	35.22	34.86
	5	**small**	**13.00**	14.22	16.36	15.00	15.00	14.71	14.71	15.00	15.14	15.00	15.28	14.86	15.00	15.07	15.00	15.14
		medium	11.07	11.72	12.64	12.14	11.86	12.22	12.14	11.21	11.14	11.07	11.29	11.14	11.21	11.07	11.21	**11.00**
		advanced	9.86	10.79	10.93	10.50	10.86	10.22	10.64	10.57	9.93	9.86	10.00	9.64	9.57	9.07	**8.93**	9.29
	10	**small**	**11.50**	13.21	13.07	13.07	12.71	12.43	13.14	12.86	12.93	12.79	12.93	12.14	12.36	12.72	12.21	11.78
		medium	9.00	9.00	9.07	9.07	8.78	8.57	8.57	8.71	8.57	8.50	8.72	**8.07**	8.22	8.21	8.14	8.21
		advanced	7.43	8.07	7.43	7.07	7.29	7.00	7.57	7.07	7.07	6.78	6.64	6.71	6.43	6.64	**6.21**	6.50
	15	**small**	10.57	11.07	11.43	10.93	10.72	10.79	10.43	10.36	**10.21**	10.64	10.64	10.64	10.36	10.36	10.36	10.50
		medium	7.50	7.64	7.21	7.00	6.50	6.64	6.93	6.64	**6.29**	6.50	6.64	6.71	6.57	6.72	6.79	6.79
		advanced	7.14	7.24	7.21	7.22	6.57	6.57	6.71	6.57	6.57	7.14	7.07	6.64	6.57	6.29	**6.14**	6.21

Table 3 Mean error rate over a 5-fold cross validation for CIFAR10 and MPEG7 datasets. T_{train} indicates the data augmentation factor during training; T_{test} indicates data augmentation factor during test. Column $T_{\text{test}} = 0$ means that no data augmentation at test is performed. In all figures, the ensemble decision is *avg*.

		$T_{\text{train}} = 0$				$T_{\text{train}} = 5$			
		orig	avg	max	mode	orig	avg	max	mode
$T_{\text{train}} = 0$	orig	-	✓	✓	✓				
	avg		-	✓	✓				
	max			-	✓				
	mode				-				
$T_{\text{train}} = 5$	orig	✓	✓	✓	✓	-			
	avg	✓	✓	✓	✓	✓	-	✓	✓
	max	✓	✓	✓	✓			-	
	mode	✓	✓	✓	✓	✓		✓	-

Table 4 Statistical significance tests with the results of the first experiment. Symbol ✓ states that the test is significantly accepted (the scheme in the row improves the scheme in the column). Significance has been set to $p < 0.05$.

strating that the approach considered is generally beneficial.

In this sense, it has been reported that the most suitable combination is a weighted vote of each prediction (*avg*) and that improvements are not obtained if the augmentation factor in test is low (less than 8 in our experiments). However, the classification error decreases in most of the cases, in a percentage dependent on the dataset and the network model used.

As a significant result, this scheme allows decreasing the error from 30.37 to 27.70 in CIFAR10 with the same neural model. The result is not striking by itself but is very interesting if considering that this scheme does not

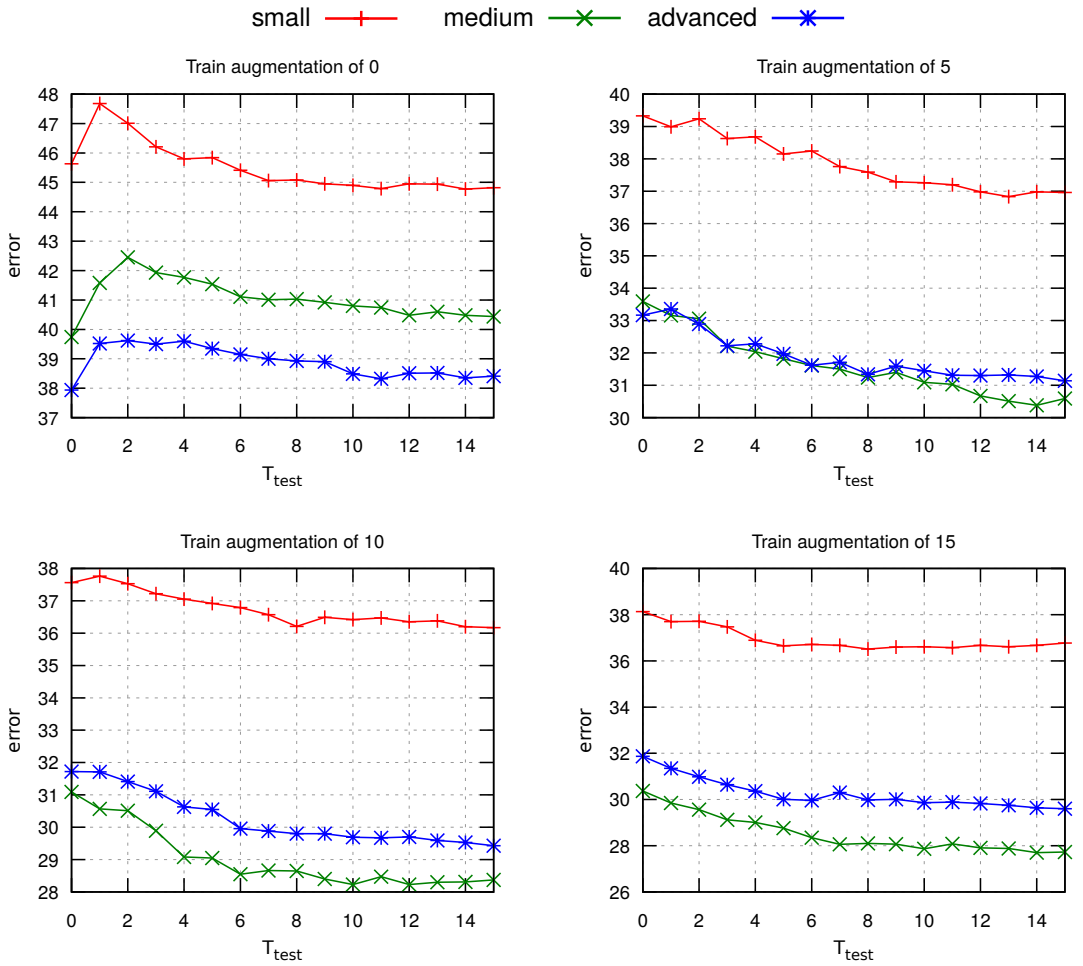


Fig. 4 Error rate obtained in CIFAR10 for each network model and data augmentation factor in training with respect to data augmentation factor in test (*avg* strategy).

require any new element, only what is available in the original scenario.

The future line of work is aimed at performing the whole process in a smart way to improve both efficiency and effectiveness. In this work, all original samples are augmented the same number of times using the same transformations. It would be interesting to infer in which cases it is worth doing test data augmentation.

Compliance with Ethical Standards

Funding: Work partially supported by the Spanish Ministerio de Ciencia, Innovación y Universidades with Juan de la Cierva - Formación grant (Ref. FJCI-2016-27873) and the Universidad de Alicante with grant GRE-16-04.

Conflict of Interest: Authors declare that they have no conflict of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Agrawal, R., Karmeshu: Perturbation scheme for online learning of features: Incremental principal component analysis. Pattern Recognition **41**(5), 1452 – 1460 (2008)
2. Ahmed, E., Jones, M., Marks, T.K.: An improved deep learning architecture for person re-identification. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
3. Aksakalli, V., Malekipirbazari, M.: Feature selection via binary simultaneous perturbation stochastic approximation. Pattern Recognition Letters **75**(Supplement C), 41 – 47 (2016)

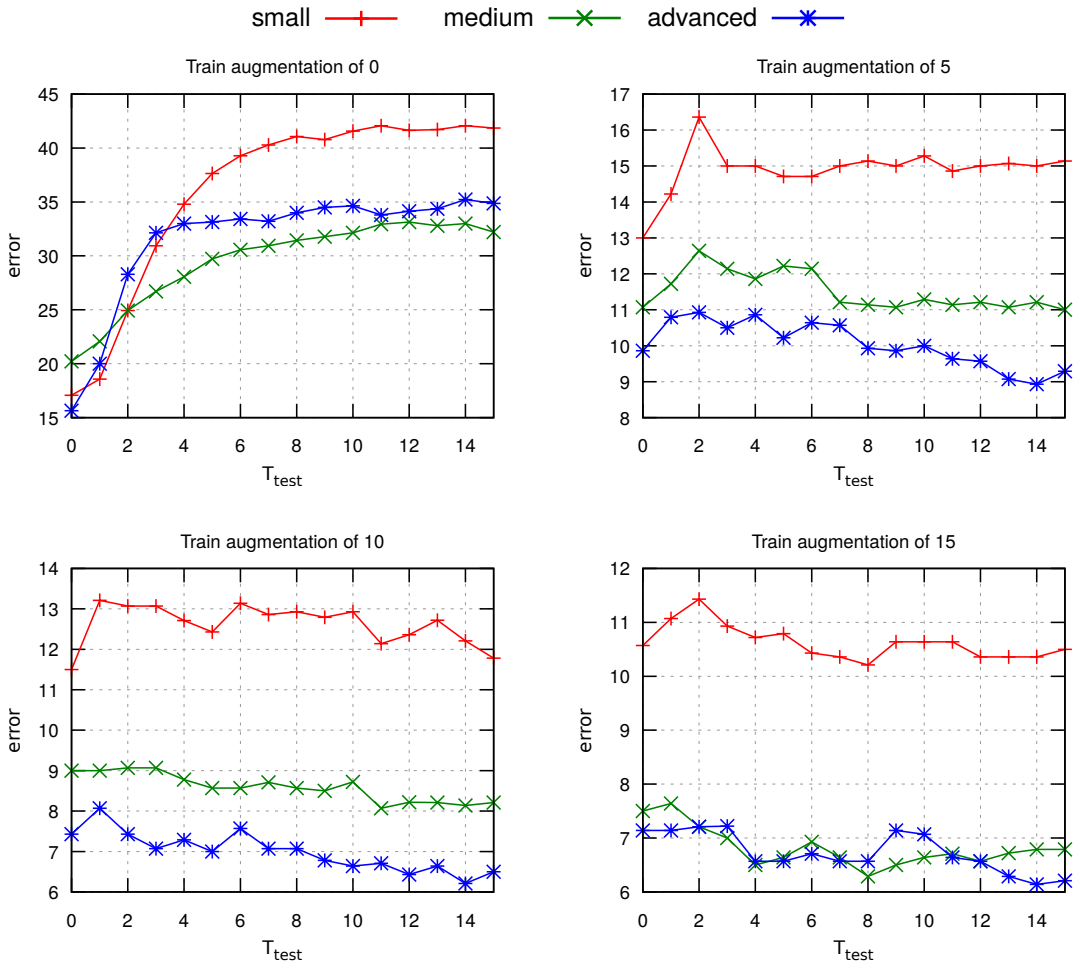


Fig. 5 Error rate obtained in MPEG7 for each network model and data augmentation factor in training with respect to data augmentation factor in test (*avg* strategy).

4. Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT'2010, pp. 177–186. Springer (2010)
5. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery **2**, 121–167 (1998)
6. Calvo-Zaragoza, J., Oncina, J.: Recognition of pen-based music notation: The HOMUS dataset. In: 22nd International Conference on Pattern Recognition, ICPR 2014, Stockholm, Sweden, August 24-28, 2014, pp. 3038–3043 (2014)
7. Cui, X., Goel, V., Kingsbury, B.: Data augmentation for deep neural network acoustic modeling. IEEE/ACM Trans. Audio, Speech and Lang. Proc. **23**(9), 1469–1477 (2015)
8. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research **7**, 1–30 (2006)
9. Duchi, J.C., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research **12**, 2121–2159 (2011)
10. Duda, R.O., Hart, P.E.: Pattern Recognition and Scene Analysis. John Wiley and Sons, New York (1973)
11. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011, pp. 315–323 (2011)
12. Goodfellow, I., Bengio, Y., Courville, A.: Regularization for deep learning. In: Deep Learning, chap. 10, pp. 228–273. MIT Press (2016)
13. Ha, T.M., Bunke, H.: Off-line, handwritten numeral recognition by perturbation method. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**(5), 535–539 (1997)
14. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. CoRR **abs/1502.01852** (2015)
15. Hull, J.J.: A database for handwritten text recognition research. IEEE Trans. on Pattern Analysis and Machine Intelligence **16**, 550–554 (1994)
16. Jain, A.K., Mao, J., Mohiuddin, K.M.: Artificial neural networks: A tutorial. Computer **29**(3), 31–44 (1996)
17. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**, 226–239 (1998)
18. Ko, T., Peddinti, V., Povey, D., Khudanpur, S.: Audio augmentation for speech recognition. In: INTER-

		$T_{\text{train}} = 0$		$T_{\text{train}} = 5$		$T_{\text{train}} = 10$		$T_{\text{train}} = 15$	
		orig	avg	orig	avg	orig	avg	orig	avg
$T_{\text{train}} = 0$	orig	-	✓						
	avg		-						
$T_{\text{train}} = 5$	orig	✓	✓	-					
	avg	✓	✓	✓	-				
$T_{\text{train}} = 10$	orig	✓	✓	✓		-			
	avg	✓	✓	✓	✓	✓	-	✓	
$T_{\text{train}} = 15$	orig	✓	✓	✓		✓		-	
	avg	✓	✓	✓	✓	✓	✓	✓	-

Table 5 Statistical significance tests with the results of the second experiment (taking the best value among the test data augmentation factors considered). Symbol ✓ states that the test is significantly accepted (the scheme in the row improves the scheme in the column). Significance has been set to $p < 0.05$.

SPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015, pp. 3586–3589 (2015)
19. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Tech. rep., University of Toronto (2009)
20. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: 26th Annual Conference on Neural Information Processing Systems, pp. 1106–1114 (2012)
21. Kuncheva, L.I.: Combining pattern classifiers : methods and algorithms. John Wiley & Sons (2004)
22. Latecki, L.J., Lakamper, R., Eckhardt, T.: Shape descriptors for non-rigid shapes with a single closed contour. In: Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on, vol. 1, pp. 424–429. IEEE (2000)
23. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
24. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
25. Lemley, J., Bazrafkan, S., Corcoran, P.: Smart augmentation learning an optimal data augmentation strategy. IEEE Access **5**, 5858–5869 (2017)
26. Lv, J.J., Cheng, C., Tian, G.D., Zhou, X.D., Zhou, X.: Landmark perturbation-based data augmentation for unconstrained face recognition. Signal Processing: Image Communication **47**, 465–475 (2016)
27. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. CoRR **abs/1312.6229** (2013)
28. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014)
29. Smith, L.N., Topin, N.: Deep convolutional neural network design patterns. arXiv preprint arXiv:1611.00847 (2016)
30. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15**(1), 1929–1958 (2014)
31. Sun, Y., Wang, X., Tang, X.: Hybrid deep learning for face verification. In: The IEEE International Conference on Computer Vision (ICCV) (2013)
32. Sutskever, I., Martens, J., Dahl, G.E., Hinton, G.E.: On the importance of initialization and momentum in deep learning. In: Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013, pp. 1139–1147 (2013)
33. Torralba, A., Fergus, R., Freeman, W.T.: 80 million tiny images: A large data set for nonparametric object and scene recognition. IEEE transactions on pattern analysis and machine intelligence **30**(11), 1958–1970 (2008)
34. Wan, L., Zeiler, M., Zhang, S., Cun, Y.L., Fergus, R.: Regularization of neural networks using dropconnect. In: S. Dasgupta, D. Mcallester (eds.) Proceedings of the 30th International Conference on Machine Learning (ICML-13), vol. 28, pp. 1058–1066. JMLR Workshop and Conference Proceedings (2013)
35. Wilkinson, R.A., Geist, J., Janet, S., Grother, P.J., et al.: The first census optical character recognition system conference. Tech. rep., US Department of Commerce (1992). DOI 10.18434/T4H01C
36. Zeiler, M.D.: ADADELTA: an adaptive learning rate method. CoRR **abs/1212.5701** (2012)
37. Zheng, W.S., Lai, J., Yuen, P.C., Li, S.Z.: Perturbation LDA: Learning the difference between the class empirical mean and its expectation. Pattern Recognition **42**(5), 764 – 779 (2009)