Data Augmentation Does Not Necessarily Beat a Smart Algorithm

Krisztian Buza[1]

Institute Jozef Stefan
Artificial Intelligence Laboratory
Jamova 39, 1000 Ljubljana, Slovenia

BioIntelligence Group
Department of Mathematics-Informatics
Sapientia Hungarian University of Transylvania
Targu Mures, Romania

buza@biointelligence.hu

Abstract: According to the "widely acknowledged truth", more training data beats algorithmic improvements in machine learning tasks. We challenge this "widely acknowledged truth" in context of data augmentation of images and recognition tasks related to images. Our observations show that real training data may be much more valuable than augmented (i.e., artificially generated) data and – most importantly – the advantage of a sophisticated algorithm relative to a simple algorithm may not be easily compensated by data augmentation.

Keywords: dynamic programming, data augmentation, machine learning, dynamic image warping

1 Introduction

State-of-the art solutions of various recognition tasks, ranging from handwriting recognition and signature verification over biometric user identification (e.g. based on the dynamic of typing) to speech recognition and image analysis tasks, are based on machine learning. Especially deep neural networks became extraordinarily popular in the last decade. Spectacular results include the detection of skin cancer [1] and retinal disease [2], "mastering the game of Go" [3], as well as recognition tasks relevant for the automotive industry [4]. Nevertheless, training deep neural networks requires a very large set of training data which is usually expensive and difficult to collect, if not impossible. For example, in case of rare diseases it may not be possible to obtain data from millions of patients. In case of biometric authentication systems, when a new user signs up to the system, the user may be asked to provide her biometric (such as handwriting or typing dynamics) a *few* times, but not thousands or millions of times.

In order to alleviate the aforementioned issues related to the collection of very large datasets, one of the most popular techniques is to generate new instances from existing instances by the (minor) modification or combination of existing instances. For example, in case of image recognition tasks, images may be shifted, elongated, resized or rotated by a few degrees, see also Fig. 2.

While data augmentation is further justified by observations showing that the prediction accuracy of machine learning techniques improves with increasing amount of training data, the actual data augmentation techniques may be somewhat ad hoc and understudied from the point of view of theory.

[1]This work was supported by the European Union through enRichMyData EU HE project under grant agreement No 101070284.

Furthermore, if sufficient training data is provided, simple algorithms have been shown to work surprisingly well in many domains[1], see e.g. nearest neighbor algorithms in time series classification [5], modified linear regression in case of drug-target interaction prediction [6] or simple classifiers in case of IT ticket text classification [7]. Moreover, Schnoebelen points out that "the widely acknowledged truth is that throwing more training data into the mix beats work on algorithms."[2]

In this paper, we will examine to which extent this "widely acknowledged truth" applies to data augmentation, one of the most prominent techniques used to improve the performance of deep neural networks. In particular, we consider an an elastic distance measure, *dynamic image warping* (DIW) which is a recent extension of dynamic time warping (DTW) for images. We compare DIW to simple distance measures, such as Euclidean and Manhattan distance. In our experiment on images of handwritten digits, DIW outperforms Euclidean and Manhattan distance even in case of data augmentation which indicates that the advantage of the more sophisticated algorithm may not be easily compensated by data augmentation.

The reminder of the paper is organized as follows: Section 2 presents dynamic image warping, while Section 3 describes our empirical observations as well as the lessons we learned from our experiments.

2 Dynamic Image Warping

Dynamic Time Warping (DTW) is an elastic distance measure for time series [8]. While comparing two time series, DTW allows for shifts and elongations. This way, DTW takes into account that, in real-world data, the same pattern is not likely to be repeated in the exactly same way. DTW is based on the paradigm of dynamic programming. When implementing DTW calculations, the entries of a matrix are filled according to a recursive rule. For more details on DTW, we refer to [9].

Dynamic Image Warping (DIW) is a recent extension of dynamic time warping (DTW) for images [10]. A digital image is a matrix of intensity values. For simplicity, we only consider grayscale images in this paper, thus, each pixel corresponds to a single intensity value. However, we note that the generalisation for color (RGB) images is straightforward. As a first step of DIW, we consider the intensity matrix of the image as a sequence of rows (or columns, respectively).

In order to compare two images, DIW compares two *sequences of sequences*. We note that in case of time series, DTW compares two *sequences of numbers* which is the major difference between DIW and DTW.

When calculating DIW, in principle, we follow the same steps as in case of DTW. The only difference between DIW and DTW is the following: while at some steps of DTW, the difference of two numbers have to be calculated, at the corresponding step in DIW, we have to calculate the distance of two sequences. In order to calculate the distance of those two sequences, we use DTW. In other words: DIW is nothing else but DTW for a sequence of sequences using DTW as inner distance.

We note that the role of columns and rows is interchangeable in case of images, therefore, when implementing our experiments, we actually calculate two DIW distances: in case of the first one, each image is considered as a sequence of rows, whereas in case of the second distance, each image is considered as a sequence of columns.

Considering an image with $N \times N$ pixels, DIW has a complexity of $\mathcal{O}(N^4)$. For this reason, we implemented DIW in Cython [11] in order to combine the efficiency of C with rapid prototyping allowed by Python. For more details see:

https://github.com/kr7/diw/blob/main/DIW.ipynb

[1] See also https://anand.typepad.com/datawocky/2008/04/data-versus-alg.html for a related discussion.

[2] https://www.datasciencecentral.com/more-data-beats-better-algorithms-by-tyler-schnoebelen/

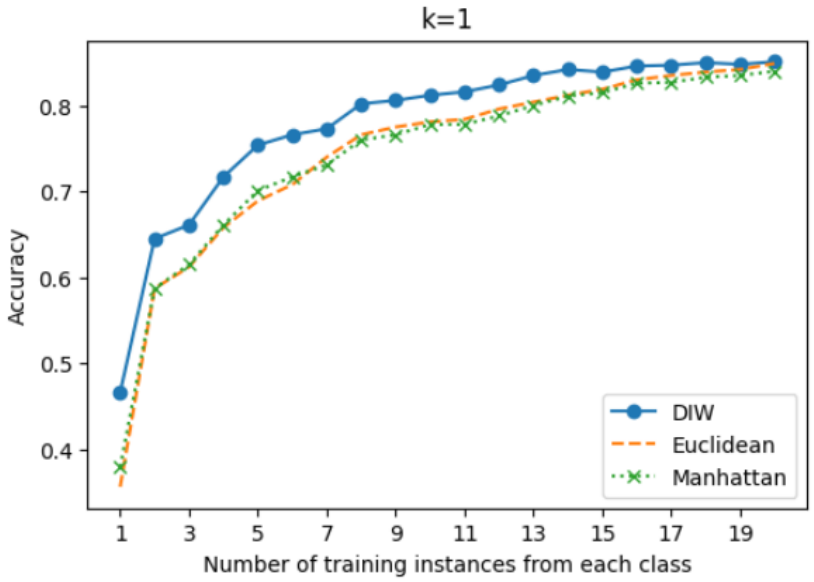

Figure 1: Classification Accuracy as Function of the Number of Training Instances

3 Experiments

Next, we present our empirical observations. In the first experiment, the classification accuracy as function of the number of training instances is studied, whereas in the second experiment, we examine the effect of data augmentation.

3.1 Classification Accuracy as Function of the Number of Training Instances

We performed experiments in the context of the recognition of handwritten digits. This is a classification task with 10 classes, where each of the classes corresponds to one of the digits '0', '1', '2', ..., '9', see also the left column of Fig. 2 for examples of images from our dataset. In our experiment, we aimed to recognize the handwritten digit using a 1-nearest neighbor classifier using either (i) DIW, or (ii) Euclidean distance or (iii) Manhattan distance to determine the nearest neighbor.

Fig. 1 shows the classification accuracy as function of the number of training instances. For example, in the case when five instances are used from each class, the training data contained five images showing a '0', another five images showing a '1', etc., thus the total size of the training data was $5 \times 10 = 50$. As one can see, the classification accuracy increases with increasing size of training data. While DIW outperforms the two other distance measures in case of few instances, the difference between the performance of the three approaches gradually decreases. When using $20 \times 10 = 200$ training instances, the classification accuracy of more than 80% is reached.

This experiment can be reproduced by running the Google colab notebook available at

https://github.com/kr7/diw/blob/main/DIW.ipynb ,

we refer to this code for further details (such as the URL of the dataset, training and test splits).

3.2 Data Augmentation

As Fig. 1 shows, in the previous experiment, compared with the case of using a single training instance per class, we observed substantial improvement in terms of classification accuracy when using more training

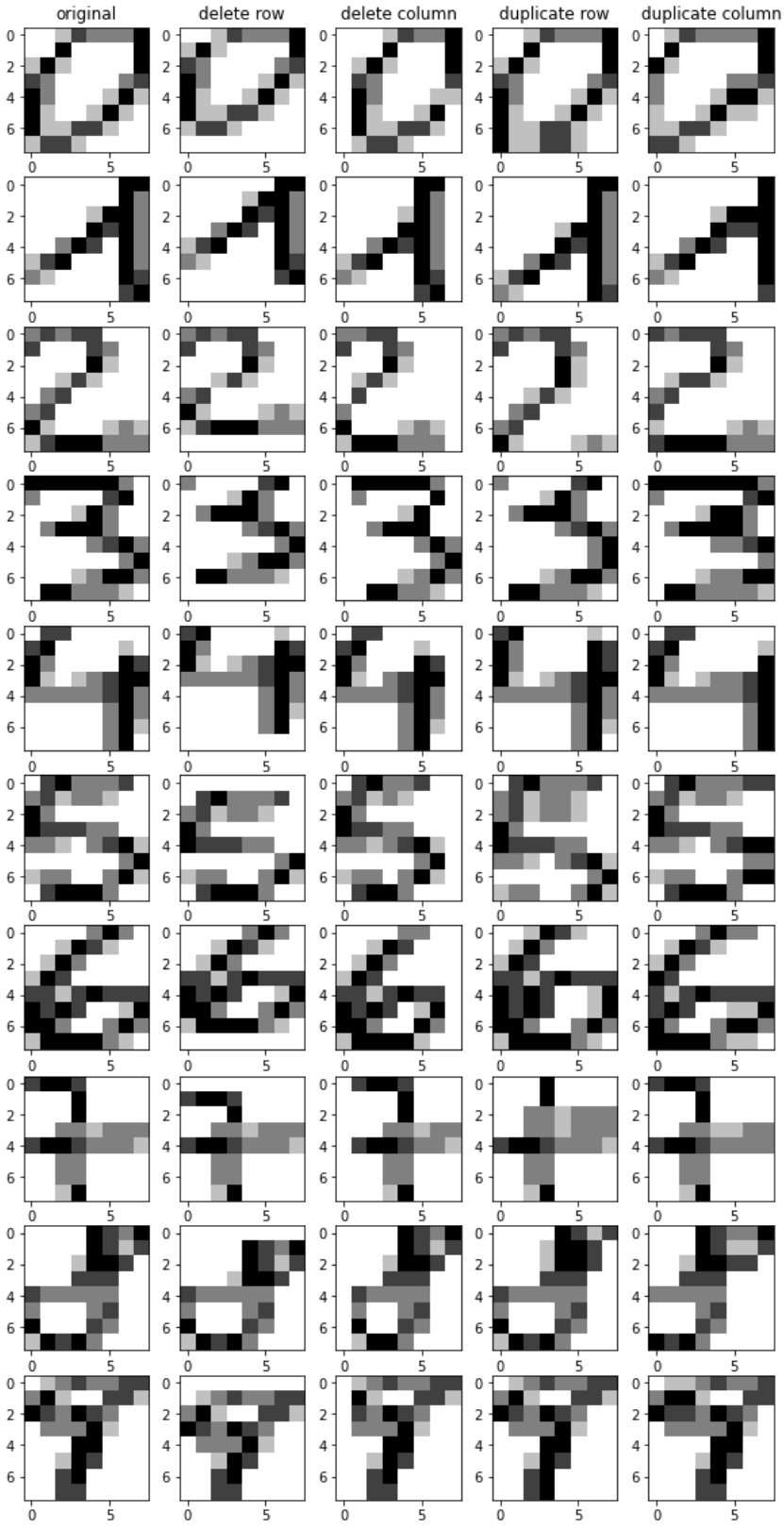

Figure 2: Data augmentation techniques used in our experiment.

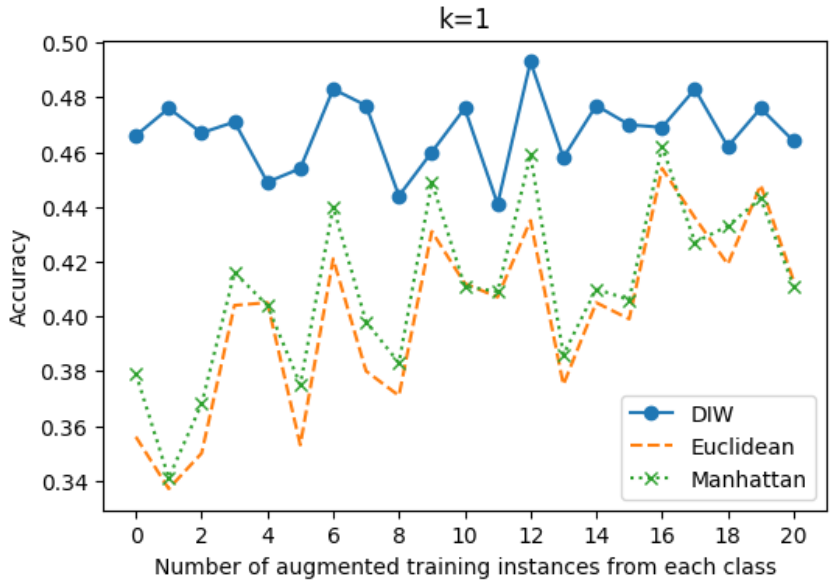

Figure 3: Classification Accuracy as Function of the Number of Training Instances

data. Next, we examine if the same improvement can be achieved using simple data augmentation techniques.

Fig. 2 shows the data augmentation techniques used in our experiment. The leftmost column shows the original image, while subsequent columns show the image after (i) deleting a randomly selected row, (ii) deleting a randomly selected column, (iii) duplication of a randomly selected row and (iv) duplication of a randomly selected column.

In this experiment, we only consider a single training instance per class. For each training instance, we create t augmented instances using the aforementioned augmentation techniques. These t augmented instances are added to the training set and the performance of the 1-nearest neighbor classifier is measured on the test set.

Fig. 3 shows the accuracy of the classifier as function of t in cases when (i) DIW, (ii) Euclidean distance and (iii) Manhattan distance was used to determine the nearest neighbor. This experiment can be reproduced by running the Google colab notebook available at

https://github.com/kr7/diw/blob/main/DIW-augmentation.ipynb .

Please see this code for further details.

Based on Fig. 3, we can make the following observations:

1. Data augmentation does not necessarily improve the performance. Is seems that data augmentation may introduce noise, although the accuracy has an increasing trend in case Euclidean and Manhattan distances.

2. More importantly, even in case of augmented data, the more sophisticated DIW algorithm beats simple distance measures which indicates that the advantage of the more sophisticated algorithm may not be easily compensated by data augmentation.

3. Last, but not least we have to note that, using data augmentation, we were not able to achieve an accuracy that is comparable with the case of using real observations as training data in Section 3.1.

An inherent limitation of our work is that we performed experiments only on a relatively small dataset from the domain of handwriting recognition which may be considered simple compared to industrial

domains, such as self-driving vehicles. While we clearly admit this limitation, we note that our setting aimed to simulate more complex domains: imagine, for example a 1 MP color (RGB) image. This image corresponds to a point in a 3 million dimensional vector space. As high-dimensional data spaces tend to be increasingly sparse, our setting, in which we only considered a single training instance when augmenting the data, aims to simulate such a sparsity.

References

[1] A. Esteva, et al., Dermatologist-level classification of skin cancer with deep neural networks, *nature* **542.7639** (2017)

[2] J. De Fauw, et al., Clinically applicable deep learning for diagnosis and referral in retinal disease, *Nature medicine* **24.9** (2018)

[3] D. Silver, et al., Mastering the game of Go with deep neural networks and tree search, *nature* **529.7587** (2016)

[4] A. Luckow, et al., Deep learning in the automotive industry: Applications and tools, *IEEE International Conference on Big Data* (2016)

[5] X. Xi, et al., Fast time series classification using numerosity reduction, *Proceedings of the 23rd international conference on Machine learning* (2006)

[6] K. Buza, L. Peška, J. Koller, Modified linear regression predicts drug-target interactions accurately, *PloS one* **15.4** (2020)

[7] A. Revina, K. Buza, V. G. Meister, IT ticket classification: the simpler, the better, *IEEE Access* **8** (2020)

[8] H. Sakoe, S. Chiba, Dynamic programming algorithm optimization for spoken word recognition *IEEE transactions on acoustics, speech, and signal processing* **26.1** (1978)

[9] K. Buza, Time series classification and its applications *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics* (2018)

[10] K. Buza, M. Antal, An Extension of Dynamic Time Warping for Images: Dynamic Image Warping, *14th Joint Conference on Mathematics and Computer Science* (2022)

[11] S. Behnel, et al, Cython: The best of both worlds, *Computing in Science & Engineering* **13.2** (2010)