MICE: Multi-layer Multi-model Images Classifier Ensemble

Plamen Angelov, *Fellow*, *IEEE* and Xiaowei Gu, *Student Member*, *IEEE*
Data Science Group,
School of Computing and Communications,
Lancaster University, Lancaster, UK
E-mail: {p.angelov, x.gu3}@lancaster.ac.uk

***Abstract*— In this paper, a new type of fast deep learning (DL) network for handwriting recognition is proposed. In contrast to the existing DL networks the proposed approach has clearly interpretable structure that is entirely data-driven and free from user- or problem-specific assumptions. It is entirely parallelizable and very efficient. First, same fundamental image transformation techniques (rotation and scaling) that are used by other existing DL methods are used to improve the generalization. The commonly used descriptors are then used to extract the global features from the training set and based on them a bank/ensemble of zero order AnYa type fuzzy rule-based (FRB) models is built through the recently introduced Autonomous Learning Multiple Model (ALMMo) method working in parallel. The final decision about the winning class label is made by a committee on the basis of the fuzzy mixture of the trained ALMMo-0 models (where "0" stands for 0 order meaning that the consequent represent a class label, a singleton, not a regression model as in the first order). The training of the proposed MICE system is very efficient and highly parallelizable. It significantly outperforms the best known methods in terms of time and is on par in terms of precision/accuracy. Critically, it offers a high level of interpretability, transparency of the classification model, full repeatability (unlike the methods that use probabilistic elements) of the results. Moreover, it allows an evolving scenario whereby the data is provided in an incremental, online manner and the system structure is being developed in parallel with the classification which opens opportunities for online and real-time applications (on a sample by sample basis). Numerical examples from the well-known handwritten digits recognition problem (MNIST) were used and the results demonstrated the very high repeatable performance after a very short training process which is in addition to the high level of interpretability, transparency.**

Keywords—deep learning, interpretability, transparency, fast training, parallelization, evolving classifiers, AnYa type fuzzy rule-based (FRB) models, Autonomous Learning Multiple Model (ALMMo) classifier.

View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Lancaster E-Prints

I. INTRODUCTION

Deep learning (DL) is now the state-of-art approach in the fields of machine learning and computer vision [1]–[6]. Using the multiple layer structure composed of linear and non-linear transformations, the deep convolutional neural networks are able to extract high-level abstractions from the data and have already demonstrated excellent results in image processing [3]–[7]. Some publications further claim that the deep artificial neural networks (ANNs) can match human performance on the recognition of handwritten digits [3], [4]. In part this success of the DL led recently to a resurrected interest to the areas of ANN and AI overall not only in science but also broader in the society [8].

However, despite the celebrated success (including the commercial one) and the increased media interest [8] from scientific point of view the DL architectures and training and design methodologies still have a number of unanswered questions and deficiencies [9], [10]:

i) There are too many *ad hoc* decisions and parameters (the number of layers, neurons, parameter values) [3]–[6];

ii) The architecture, the feature extraction and training process are opaque (very low human interpretability) [1], [4], [6];

iii) The training process requires a large amount of time and resources [3]–[5] which preclude training and adaptation in real time;

iv) The training process is not parallelizable [3]–[6].

These deficiencies make DL a tedious and heavy methodology known to specialists and experts and hinder the wider acceptability.

In this paper, we propose a new method that has a potential to democratize DL methodology due to great progress in addressing the above concerns plus the ability to develop dynamically evolving DL architectures that learn and are being trained online on a sample by sample basis. Furthermore, it is fully parallelizable, transparent (no *ad hoc* decisions, no user- or problem-specific parameters. Yet, it is as precise as the best DL architectures reported and much [illegible] weights and coefficients [illegible].

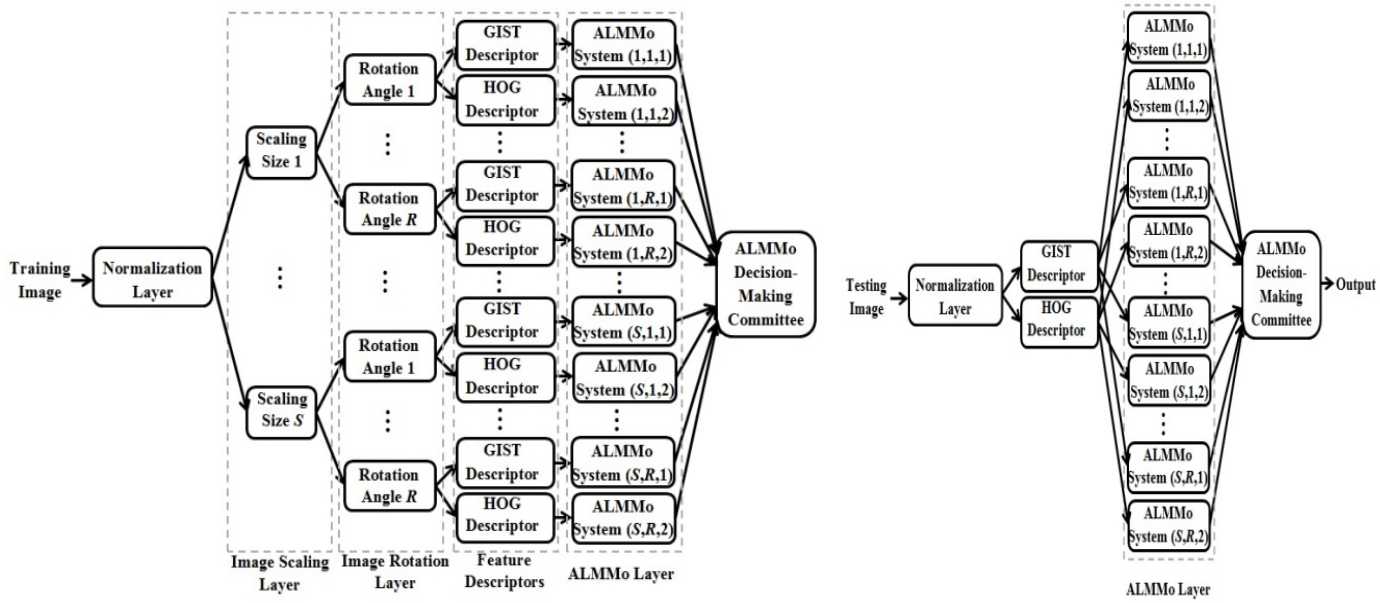
The proposed approach consists of a number of components involving:

i) Image transformation layers involving image normalization and affine distortions operations (same as the one used in [4];

ii) Global feature descriptors (GIST [11] and HOG [12]);

iii) Zero order AnYa type FRB [23] trained using the recent Autonomous Learning Multiple Model (ALMMo) method [13], [14] ;

This work was supported by The Royal Society (Grant number IE141329/2014)



(a) The structure of learning process (b) The structure of validation process

Fig. 1. The proposed apporach

iv) A decision-making committee for the final class outcome following the ensemble/bank of parallel ALMMo.

The proposed deep network employs the most widely used image transformation techniques and feature descriptors without handcrafting complex and opaque elements of the architecture (e.g. the number of hidden layers, number of neurons in the hidden layers, their function, etc.). The backbone of the proposed architecture is the bank of AnYa type IF … THEN rules [23]. The AnYa FRB classifier is designed using the recently introduced ALMMo [13], [14] within the Empirical Data Analytics (EDA) framework [15] which makes the learning process entirely data-driven and highly parallelizable. The proposed architecture is simple and transparent/interpretable. The training is very fast due to the non-parametric and non-iterative nature of AnYa and ALMMo.

The numerical example clearly demonstrates that the proposed approach is able to perform highly accurate handwriting recognition after a very short and transparent learning process.

The remainder of the paper is organized as follows. In sections II-VI we describe the proposed overall network, image transformation techniques, feature descriptors, ALMMo system as well as the numerical experiment and analysis. The paper is concluded by section VII.

II. Problem Definition and the Proposed Approach

The proposed architecture structure of the proposed approach is presented in Fig. 1(a) for the training stage, where S is the number of different scaling sizes; R is the number of different rotation angles (same as the one used in [4]). The triplet (i, j, k) where $i = 1, 2, ..., S$, $j = 1, 2, ..., R$ and $k = 1, 2$ can be used as an ID of a given ALMMo.

During the training stage the proposed network same as [4] is using the most fundamental image transformation techniques (scaling and rotation) which leads to a significant increase of the amount of the training data samples and, respectively, improves significantly the generalization capabilities of the classification [4]. The images that resulted from rotation and scaling of the original images are then sent to different learning engines as illustrated in Fig. 1. After scaling and normalization the well-known image feature descriptors (histogram of oriented gradients or HOG and GIST) are used to extract the global features from those training sets. The zero order AnYa type FRB classifiers are then trained using ALMMo in parallel based on the different features from different training sets.

The training is non-iterative and non-parametric and is, therefore, very fast. Once it is finished for a given set of data samples, the trained ALMMo are used in as an ensemble to form the committee-based decision for validation, see Fig. 1(b). During the validation process, the HOG and GIST features are extracted from each validation image and used by the corresponding trained AnYa type FRB classifier which further feeds into the decision-making committee to produce the final recognition result accordingly, Fig. 1 (b).

The proposed multi-layer multi-model structure is described in more detail in the following sections.

III. Image Transformation

In this section, we will briefly describe and compare the image transformation techniques including normalization, affine distortion and elastic distortion commonly used in the deep convolutional neural networks for handwriting recognition [3], [4], [7]. Although, these pre-processing transformations are generally applicable to various image processing problems (same as our newly proposed approach) in this paper we will focus on a specific problem od so called MNIST dataset [16], [17]. This is a well-known benchmark

image processing problem used back in 1980s by Fukunaga for his famous Neucognitron, perhaps one of the first DL architecture [18]. It is a complex problem and concerns handwriting digits recognition which contains 60000 training and 10000 testing greyscale images. The total of 70000 images are of 10 classes (equally distributed in groups of 6000 training images and 1000 testing images corresponding to each of the 10 digits (from "0" to "9"), see Figs 2-4. It is also important to stress that all images are unique (there are no pairs of same images, no repetition of the same image). The images are centered in a 28×28 box [16].

Many approaches have been proposed and reported with the best result published to the moment provides a recognition accuracy of 99.77% [4]. However, in this work there is one additional pre-processing, image transformation technique used, namely the elastic distortion (as described in section III) which is not only opaque (not clearly reported in [4] but also is random in nature. This combined with the other random elements of the architecture leads to the results being different each time training is performed on the same data which not only requires cross-validation, but also places a shadow and questions over the value of 99.77% - can it be repeated, guaranteed etc.

In the method proposed in this paper, we only use normalization and affine distortion (scaling and rotation) techniques as the three pre-processing layers of this network. The results are fully repeatable and no randomness is involved at any stage.

A. Image Normalization

Normalization is a common process in image processing that changes the value range of the pixels within the image. The goal is to transform the image such that the values of pixels are mapped into a more familiar or normal range. This operation can be used to readjust the degree of illumination of the images.

In the proposed approach, the most commonly used linear normalization is used to fit the original pixel value range of [0, 255] into the range of [0,1].

B. Affine Distortions

Affine distortion can be done by applying affine displacement fields to images, computing for every pixel a new target location with respect to the original one. Affine distortions including rotations and scaling are very effective to improve the generalization and decrease the overfitting [3], [4], [7].

i. Image scaling

Image scaling refers to the resampling and resizing of a digital image [19], [20]. There are two types of image scaling: *i)* image contraction and *ii)* image expansion. Image scaling is achieved by using an interpolation function. There is a number of different interpolation methods for image resizing reported in the literature [19]–[22], e.g. nearest neighbor interpolation, bilinear interpolation and bicubic interpolation methods. In this paper, we use the most commonly used bicubic interpolation method [21], [22] which considers the nearest 16 pixels (4x4) in the neighborhood and calculates the output pixel value as their weighted average. Since the 16 neighboring pixels are at various distances from the output pixel, closer pixels are given a higher weighting in the calculation.

The specific image scaling operation used in the proposed approach is more often called "width normalization" [3], [4], [7]. In the proposed approach, we resize the original training images from their original dimension of 28×28 into 7 ($S=7$) different sizes, namely: *i)* 28×22 ; *ii)* 28×24 ; *iii)* 28×26 ; *iv)* 28×28 ; *v)* 28×30 ; *vi)* 28×32 , and *vii)* 28×34 . Then, for the first 3 types of resized images that are smaller than the original size (*i)* to *iii)*) we use white pixels to fill the blank spaces so that we maintain the original size of 28×28 . For the last 3 types of resized images that are larger than the original size (*v)* to *vii)*) we cut the redundant parts in both sides of the images to restore their original sizes while still keeping the digits in the center of the original size (28×28) frame. As a result, we create 7 new training sets from the original one (in the MNIST case this makes 7 times 60000 = 420000 training images) [4].

ii. Image Rotation

Image rotation is a common image pre-processing technique performed by rotating an image at certain angle around the center point [23]. Usually, the nearest neighbor interpolation is used after the rotation and the values of pixels that are outside the rotated images are set to *0* (black).

In the proposed approach, all the training images (the original as well as the scaled ones; so, in total, 420000 images) were rotated at different angles in a counter clockwise direction starting from *-15* degrees going through *0* (no rotation) up to *15* degrees with an interval of 3 degrees. After the rotation, 11 ($R=11$) different rotated images are obtained from each image that is being rotated, Fig.3. Therefore, as a result of all pre-processing we apply in this paper for the MNSIT problem 4.62 million images of size 28 by 28 pixels are being generated. This process is similar to the one used by other methods [4], but we do not use any random elements such as the elastic distortion (Fig. 4) which also creates further new images.

C. Elastic Distortion

Elastic distortion is a more advanced and effective technique to expand the dataset and improve the generalization [3], [4], [7]. The elastic distortion is done by first generating random displacement fields and then convolving the displacement fields with a Gaussian function of standard deviation σ (in pixels) [7]. This type of image deformation has been widely used in the state-of-art deep convolutional neural networks for handwriting recognition [3], [4] and largely improved the recognition accuracy.

However, this kind of distortion exhibits a significant randomness which puts in question the achieved results, repeatability and requires a cross-validation which further obstructs the online applications and the reliability of the results. In addition, it adds user-specific parameters that can be chosen differently. For a particular image, each time the

elastic distortion is performed, an entirely new image is being generated.

Moreover, by repeating the elastic distortion operation on a few training images of a certain class one can actually obtain all the training images and testing images of the same class from this image. This puts the question of fairness because in a real life situation the validation data will not be known and available at all, but in scientific research and in project work before designing a tool one can demonstrate very strong performance but the validation data (in this case the 10000 images) are actually known/available. Unless there is repeatability to allow the experiments to be performed again by others and to get exactly the same results reporting results that include random steps and choices undermines the strength of the reported result. Therefore, in the proposed approach we decided not to include elastic deformation and any elements that have random element and use instead only the affine distortions to ensure the repeatability.

Fig. 4 presents an example of the distorted images using the same elastic distortion as in [3], [4], one can see that, every elastic distorted image is different even if they come from the same original image. Therefore, the randomness of elastic distortion may largely affect the reproducibility of the proposed approach.

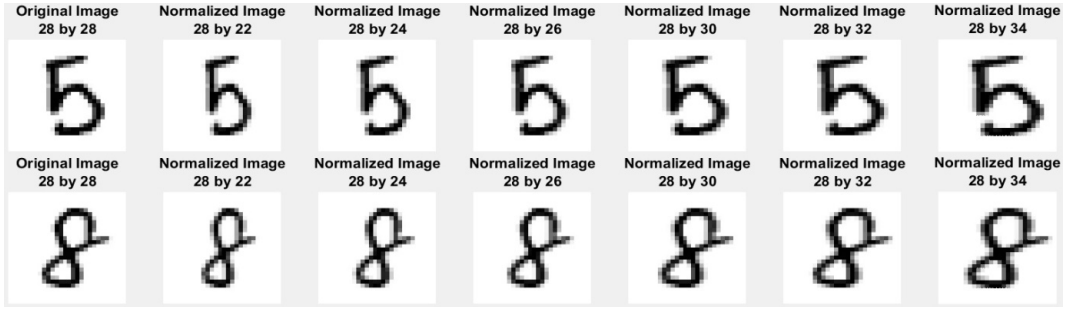


Fig. 2. Illustrative examples of image rotation

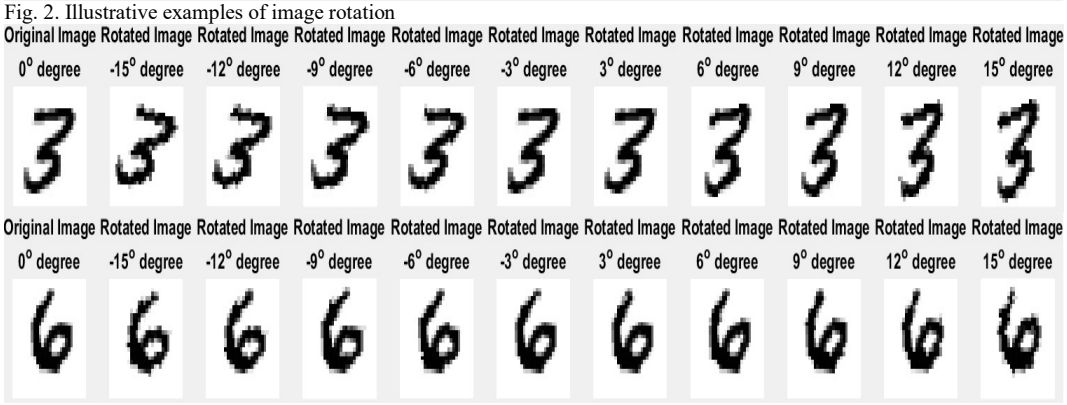


Fig. 3. Illustrative examples of width normalization

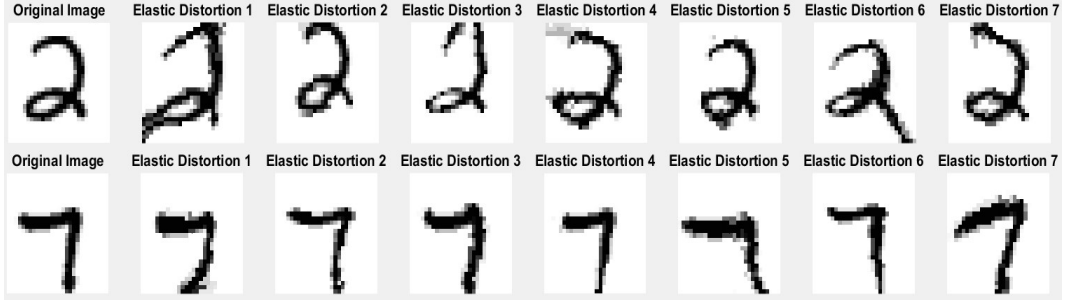


Fig. 4. Illustrative examples of elastic distortion

IV. Image Feature Extraction

Feature extraction is very important to solve computer vision problems such as object recognition, content-based image classification and image retrieval [24]. The extracted features has to be informative, non-redundant, and, most importantly, to be able to facilitate the subsequent learning and generalization. In this section, we will focus on the widely used GIST [11] and HOG [12] descriptors.

A. GIST descriptor

The GIST descriptor, as firstly introduced in [11], extracts the global features of images and gives an impoverished and coarse version of the principal contours and textures of the image. The fundament of the GIST features is the Gabor filter [11]. It is computationally efficient and there is no interdependence between the GIST features of different images, which allows large-scale parallelization. Once the GIST feature of an image is extracted and stored, there is no need to repeat the same feature extraction process on the same image any more. In the proposed approach, we use the same GIST descriptor without any modifications as described in [11]. A 1×512 dimensional vector of GIST feature denoted as $\boldsymbol{g}=[g_1,g_2,...,g_{512}]$ has been extracted from each image.

B. HOG descriptor

HOG descriptor [12], [25] has been proven to be very successful in various computer vision tasks such as object detection, texture analysis and image classification. The HOG descriptor decomposes an image into small cells, computes a histogram of oriented gradients in each cell, normalizes the result using a block-wise pattern, and the descriptor is the concatenation of these histograms [12]. Similar to the GIST feature, the task of HOG features extraction from the dataset can be parallelized to improve the computational efficiency and once the HOG feature of an image is extracted and stored, there is no need

to extract it again.

In the proposed approach, as the size of the images is 28×28, a 4×4 size patch is used for computing the HOG feature. Thus, for each image, a 1×1296 dimensional vector of HOG feature denoted as $\boldsymbol{h}=[h_1,h_2,...,h_{1296}]$, can be extracted. To improve the distinctiveness of the HOG features, we also employ a non-linear mapping function which expands the range of values of the HOG feature [10]:

$$\kappa(x)=\mathrm{sgn}(x)\left[\exp\left[\left(1+\mathrm{sgn}(x)x\right)^2\right]-\exp(1)\right] \quad (1)$$

where $\mathrm{sgn}(x)=\begin{cases}1, & x>0\\ 0, & x=0\\ -1, & x<0\end{cases}$.

The nonlinearly mapped HOG feature:

$$\boldsymbol{h}\leftarrow\left[\kappa(h_1),\kappa(h_2),...,\kappa(h_{1296})\right] \quad (2)$$

is further used for training the proposed multi-layer multi-model DL classifier composed of parallel AnYa/ALMMo.

V. AnYa FRB and ALMMo Ensemble

The Autonomous Learning Multiple Model (ALMMo) [13], [14] is a recently introduced autonomous learning method based on AnYa type FRB system [26] within the Empirical Data Analytics (EDA) framework [15]. It has the following form:

$$IF\ (Image\ \sim\ Prototype)\ THEN\ (Class) \quad (3)$$

where "~" denotes similarity which can be also seen as a fuzzy degree of satisfaction/membership [26] or typicality [15], [27].

ALMMo is a method to autonomously extract the nonparametric shape-free data clouds [26] and form simple linguistic rules [26] from the empirical observations. Its multi-model structure is identified in a fully data-driven way without making prior assumptions about the data distributions and number of local models. The meta-parameters of the ALMMo are derived from the data in a recursive way, which makes this multi-model system very suitable for streaming data processing.

ALMMo can be used for prediction [28], but also for classification. ALMMo classifiers (same as other FRB classifiers [29]) can be of 0, 1 or higher order. The meaning is the order of the consequent (THEN) part of the rules which can be a singleton, integer constant which directly represents the class label (0 order), linear regression over the features (1 order), a Gaussian or another non-linear higher order output. In this paper we use 0 order ALMMo classifier [14]. An ALMMo-0 contains C individual classifiers corresponding to different classes that the dataset has, where C denotes the number of classes (in this paper, $C=10$, one for each digit). The advantages of the ALMMo-0 classifier are as follows:

i) The learning process is fast, non-iterative, non-parametric and entirely data-driven;

ii) The classification model is transparent and human understandable (IF…THEN rules);

iii) The system structure can be dynamically evolving;

iv) The multiple model structure is convenient for parallelization reducing the computation load.

A. ALMMo-0 System Identification

As it is shown in Fig. 1(a), for each training set, two ALMMo-0 systems based on the HOG and GIST features are developed automatically from the data. Therefore, in total $2\cdot S\cdot R$ ALMMo-0 systems are trained. Since there is no interdependence existing between the images from different classes or different trainings sets, one can train the C sub-systems of $2\cdot S\cdot R$ systems in parallel (for the MNIST problem that makes *154*). Therefore, the learning process can be divided into $2\cdot S\cdot R\cdot C$ independent tasks (for the MNIST problem this makes *1540*), which can largely improve the computation efficiency.

The learning process of a particular ALMMo-0 system is summarized and given in the form of pseudo code as follows [14].

The learning process of the ALMMo-0 system

While the HOG/GIST feature, $\boldsymbol{x}_k^i$, is extracted from a new image of the i^{th} class

i. $\boldsymbol{x}_k^i\leftarrow\dfrac{\boldsymbol{x}_k^i}{\left\|\boldsymbol{x}_k^i\right\|}$

($\left\|\boldsymbol{x}_k^i\right\|$ is the Euclidean norm of $\boldsymbol{x}_k^i$)

ii. **If** ($k=1$) **Then**

1. $\boldsymbol{\mu}_1^i\leftarrow\boldsymbol{x}_1^i$
   ($\boldsymbol{\mu}_1^i$ is the global mean);
2. $F^i\leftarrow1$
   (F^i is the number of fuzzy rules/ data clouds)
3. $\boldsymbol{x}_1^{*i}\leftarrow\boldsymbol{x}_1^i$
   ($\boldsymbol{x}_1^{*i}$ is the focal point of the first data cloud)
4. $M_1^{*i}\leftarrow1$
   (M_1^{*i} is its corresponding support)
5. $r_1^{*i}\leftarrow r_o$
   (r_1^{*i} is its radius, $r_o=\sqrt{2\left(1-\cos(30^o)\right)}$)

iii. **Else**

1. Update the global mean, $\boldsymbol{\mu}_{k-1}^i$ using equation (4):

$$\boldsymbol{\mu}_k^i\leftarrow\frac{k-1}{k}\boldsymbol{\mu}_{k-1}^i+\frac{1}{k}\boldsymbol{x}_k^i \quad (4)$$

2. Calculate the density, D of $\boldsymbol{x}_k^i$ using equation (5):

$$D_k\left(\boldsymbol{x}_k^i\right)=\frac{1}{1+\frac{\left\|\boldsymbol{x}_k^i-\boldsymbol{\mu}_k^i\right\|^2}{1-\left\|\boldsymbol{\mu}_k^i\right\|^2}} \quad (5)$$

3. Update the density of the existing focal points $\boldsymbol{x}_j^{*i}$ ($j=1,2,...,F^i$) using equation (5) and obtain $D_k\left(\boldsymbol{x}_j^{*i}\right)$ ($j=1,2,...,F^i$);

4. **If** $\left(D_k\left(\boldsymbol{x}_k^i\right)>\max_{j=1,2,...,F_i}\left(D_k\left(\boldsymbol{x}_j^{*i}\right)\right)\right)$ **Or** $\left(D_k\left(\boldsymbol{x}_k^i\right)<\min_{j=1,2,...,F_i}\left(D_k\left(\boldsymbol{x}_j^{*i}\right)\right)\right)$ **Then**

- $F^i \leftarrow F^i+1$;
- $\boldsymbol{x}_{F^i}^{*i} \leftarrow \boldsymbol{x}_k^i$;
- $M_{F^i}^{*i} \leftarrow 1$;
- $r_{F^i}^{*i} \leftarrow r_o$;

5.**Else**

- Find the nearest data cloud using equation (6):

$$\boldsymbol{x}_{nearest}^{*i}=\arg\min_{j=1,2,...,F_i}\left(\left\|\boldsymbol{x}_k^i-\boldsymbol{x}_j^{*i}\right\|\right) \quad (6)$$

- **If** ($D_k\left(\boldsymbol{x}_k^i\right)>\max_{j=1,2,...,F_i}\left(D_k\left(\boldsymbol{x}_j^{*i}\right)\right)$) **Then**

  * $\boldsymbol{x}_{nearest}^{*i} \leftarrow \frac{M_{F^i}^{*i}}{M_{F^i}^{*i}+1}\boldsymbol{x}_{nearest}^{*i}+\frac{1}{M_{F^i}^{*i}+1}\boldsymbol{x}_k^i$;
  * $M_{nearest}^{*i} \leftarrow M_{nearest}^{*i}+1$;
  * $r_{nearest}^{*i} \leftarrow \sqrt{0.5\left(\left(r_{nearest}^{*i}\right)^2+\left(1+\left\|\boldsymbol{x}_{nearest}^{*i}\right\|^2\right)\right)}$;

- **Else**

  * $F^i \leftarrow F^i+1$;
  * $\boldsymbol{x}_{F^i}^{*i} \leftarrow \boldsymbol{x}_k^i$;
  * $M_{F^i}^{*i} \leftarrow 1$;
  * $r_{F^i}^{*i} \leftarrow r_o$;

- **End If**

6. **End If**

iv. **End If**

End While

B. Classification by an individual ALMMo system

The output of a single ALMMo-0 system is based on the "winner takes all" principle in regards to the number of AnYa rules that form a particular ALMMo classifier [14]. Each ALMMo-0 systems consists of C independent sub-systems corresponding to the C classes of the data. Therefore, C scores of confidence denoted as $\boldsymbol{\lambda}_i(\boldsymbol{x})=\left[\lambda_{i,1}(\boldsymbol{x}),\ \lambda_{i,2}(\boldsymbol{x}),\ ..,\lambda_{i,C}(\boldsymbol{x})\right]$. The confidence can be obtained from a particular ALMMo-0 system for a specific validation image as follows:

$$\lambda_{i,j}(\boldsymbol{x})=\arg\max_{k=1,2,...,F^{i,j}}\left[\exp\left[-\frac{\left\|\boldsymbol{x}-\boldsymbol{x}_k^{*i,j}\right\|^2}{2}\right]\right] \quad (7)$$

where $\boldsymbol{x}$ is the global feature of the validation image (GIST or HOG), $\boldsymbol{x}=\frac{\boldsymbol{h}}{\|\boldsymbol{h}\|}$ *or* $\frac{\boldsymbol{g}}{\|\boldsymbol{g}\|}$; i is the index of the ALMMo-0 system, $i=1,2,...,SR$; j is the index of the sub-system, $j=1,2,...,C$; $\boldsymbol{x}_k^{*i,j}$ is the k^{th} focal point of the sub system and $F^{i,j}$ is the number of focal points within this sub-system.

C. Decision-Making Committee

Since there are $2\cdot S\cdot R$ ALMMo-0 systems identified from the expanded training sets in the training stage, it is of critical importance how to combine the partial $2\cdot S\cdot R$ outputs. In this paper, we use a decision-making committee. One possible way for the ensemble of ALMMo-0 systems to make decisions is voting. However, the voting mechanism ignores the majority of the information gathering together from the committee.

In the proposed approach, we firstly divide the individual ALMMo classifiers into two groups, namely, HOG and GIST based on the global features they used. This is due to the fact that different features have different norms and as a result the value range of the scores of confidence based on different features are not comparable.

After all the committee members give their outputs, we can get two sets of confidence score vectors: $\{\boldsymbol{\lambda}(\boldsymbol{g})\}$ and $\{\boldsymbol{\lambda}(\boldsymbol{h})\}$, where each set contains SR elements. Based on $\{\boldsymbol{\lambda}(\boldsymbol{g})\}$ and $\{\boldsymbol{\lambda}(\boldsymbol{h})\}$, an overall score of confidence for each class can be obtained using equation (8). As one can see, the overall confidence score takes two types of global features as inputs, the overall confidence scores and the maximum confidence scores into consideration. Thus, it integrates the most important information to make the judgement by differ from the simple voting mechanism as used in many other works.

$$\begin{aligned}\Lambda_j=&\frac{1}{2}\left(\frac{1}{SR}\sum_{i=1}^{SR}\lambda_{i,j}\left(\frac{\boldsymbol{g}}{\|\boldsymbol{g}\|}\right)+\max_{i=1,2,...,SR}\left(\lambda_{i,j}\left(\frac{\boldsymbol{g}}{\|\boldsymbol{g}\|}\right)\right)\right)\\&+\frac{1}{2}\left(\frac{1}{SR}\sum_{i=1}^{SR}\lambda_{i,j}\left(\frac{\boldsymbol{h}}{\|\boldsymbol{h}\|}\right)+\max_{i=1,2,...,SR}\left(\lambda_{i,j}\left(\frac{\boldsymbol{h}}{\|\boldsymbol{h}\|}\right)\right)\right)\end{aligned} \quad (8)$$

The overall final classification decision is made using the "winner takes all" principle as follows:

$$Label=\arg\max_{i=1,2,...,C}\left(\Lambda_j\right) \quad (9)$$

TABLE I. COMPARISON BETWEEN THE PROPOSED APPROACH AND DIFFERENT DNN APPROACHES

Approaches	Accuracy	Training Time	PC Parameters	GPU Used	Elastic Distortion	Tuned Parameters	Iteration	Reproducibility	Parallelization	Evolving Ability
The Proposed MICE approach (this paper)	99.32%	*i.* Less than 1 minute per class for each member of the ALMMo committee with GIST features; *ii.* Less than 4 minutes per class for each member of the ALMMo committee with HOG features.	Core i7-4790 (3.60GHz), 16 GB DDR3	None	NO	NO	NO	YES	YES	YES
Committee of 7 Convolutional Neural Networks [3]	99.73% ± 2%	Almost 14 hours for each one of the 5 DNNs.	Core i7-920 (2.66GHz), 12 GB DDR3	2 × GTX 480 & 2 × GTX 580	YES	YES	YES	NO	NO	NO
Committee of 35 Convolutional Neural Networks [4]	99.77%	Almost 14 hours for each one of the 35 DNNs.	Core i7-920 (2.66GHz), 12 GB DDR3	2 × GTX 480 & 2 × GTX 580	YES	YES	YES	NO	NO	NO

VI. EXPERIMENT DEMONSTRATION AND DISCUSSION

In this section, we will demonstrate the experiments conducted based on the MNIST dataset and make a comparison with the best reported state-of-art approaches.

In the proposed MICE approach, as described above we expanded the original dataset into *77* different training sets using scaling and rotation. With the HOG and GIST features extracted from each training set, we trained *154* ALMMo-0 systems in parallel for each digit. The accuracy our MICE classifier that contains a set of fully transparent and interpretable fuzzy IF…THEN…rules of AnYa type [26] is 99.32% (only 68 errors for the 10000 validation images). We also compared our approach with the best reported state-of-art techniques in terms of accuracy, time and complexity, reproducibility, parallelization, evolving capability and presented it in Table I.

The training time required by the ALMMo-0 systems based on the HOG and GIST features of the original training set are tabulated in Table II. However, it has to be noticed that these are indicative times and the exact training time is difficult to be provided because the amount of training time required by the ALMMo-0 systems based HOG and GIST features for the same training set are different. In addition, the amount of training time required by the ALMMo-0 systems based on the same type of features of different training sets also vary, though slightly. The time consumptions are measured on MATLAB R2015a platform using a PC with dual core i7 processor with clock frequency 3.6GHz each, 16GB RAM and Window 10 operation systems.

From Table II we can see, the maximum training time for an ALMMo-0 sub-system is 222.73 seconds (less than 4 minutes). As each part of the proposed approach is highly parallelizable, with unlimited computational resources, the whole training process can be finished well within 5 minutes for all 60000 original training images. Furthermore, because of the inherent ability of the ALMMo to be recursively updated and dynamically to evolve an incremental online scenario whereby images are provided one by one can also be realized and will be a matter of our future work. This opens possibilities for fast, real-time, online, highly precise self-learning deep interpretable classifiers to be autonomously build

VII. CONCLUSION

In this paper, we proposed a novel fast, non-parametric DL ensemble rule-based classifier and applied it to a well- known benchmark problem of handwritten digits recognition. The core of the proposed approach is the ensemble/bank of parallel AnYa type FRB classifiers and their learning using the

TABLE II. TIME CONSUMPTION FOR THE LEARNING PROCESS PER SUB-SYSTEM (IN SECONDS)

Sub-system #		1	2	3	4	5	6	7	8	9	10
Digital		"0"	"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"
Feature	GIST	44.09	36.73	46.51	49.69	41.66	38.95	42.11	41.66	47.85	43.12
	HOG	194.94	167.44	204.18	222.73	194.17	169.23	179.67	192.41	191.04	183.49

recently introduced ALMMo-0 method within the recently introduced EDA framework. The proposed approach only involves the most fundamental image transformation techniques to improve the generalization and uses the commonly used feature descriptors (GIST and HOG) for its learning process. With a number of ALMMo-0 systems trained in parallel as the learning engines, a decision-making committee is organized for producing the overall classification output. The experimental results demonstrate the excellent performance of the proposed approach outperforming the best reported methods by providing highly interpretable and transparent, repeatable, parallelizable, potentially evolving, much faster and on par in terms of precision results.

Compared with the state-of-art deep convolution neural networks giving the current best recognition results on the MNIST dataset, the proposed approach has the following advantages:

i) The learning process is fast, non-parametric and non-iterative;

ii) The learning process is transparent and human understandable;

iii) The learning process is highly parallelizable;

iv) The structure of the network is evolving in its nature;

v) The results are reproducible (no randomness is involved and no *ad hoc* decisions about the architecture of the classifier).

As future work, we will further involve elastic distortion to our approach aiming at achieving even higher accuracy results. We will experiment with the evolving aspect of the proposed approach and will apply the proposed approach to various problems including human action recognition, remote sensing, etc.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nat. Methods*, vol. 13, no. 1, pp. 35–35, 2015.

[2] T. Kuremoto, S. Kimura, K. Kobayashi, and M. Obayashi, "Time series forecasting using a deep belief network with restricted Boltzmann machines," *Neurocomputing*, vol. 137, pp. 47–56, 2014.

[3] D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Convolutional neural network committees for handwritten character classification," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2011, vol. 10, pp. 1135–1139.

[4] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column Deep Neural Networks for Image Classification," in *Cvpr*, 2012, pp. 3642–3649.

[5] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *Int. Conf. Learn. Represent.*, pp. 1–14, 2015.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Adv. Neural Inf. Process. Syst.*, pp. 1–9, 2012.

[7] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," *Doc. Anal. Recognition, 2003. Proceedings. Seventh Int. Conf.*, pp. 958–963, 2003.

[8] V. Mnih, K. Kavukcuoglu, D. Silver, A. a Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[9] J. Bruna and S. Mallat, "Invariant Scattering Convolution Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1872–1886, 2013.

[10] P. Angelov, X. Gu, and J. Principe, "Fast Feedforward Non-parametric Deep Learning Network with Automatic Feature Extraction," in *International Joint Conference on Neural Networks*, 2017, *to appear.*

[11] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.

[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Proc. - 2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, CVPR 2005*, vol. I, pp. 886–893, 2005.

[13] P. P. Angelov, X. Gu, and J. C. Principe, "Autonomous Learning Multi-model Systems from Data Streams," *IEEE Trans. Fuzzy Syst.*, *submitted* 2016.

[14] P. P. Angelov and X. Gu, "Autonomous Learning Multi-Model Classifier of 0- Order (ALMMo-0)," in *IEEE International Conference on Evolving and Autonomous Intelligent Systems*, 2017, *submitted*.

[15] P. Angelov, X. Gu, and D. Kangin, "Empirical data analytics," *Int. J. Intell. Syst.*, *to appear,* 2016.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.

[17] "MNIST Dataset," *http://yann.lecun.com/exdb/mnist/*.

[18] K. Fukushima, "Neocognitron for handwritten digit recognition," *Neurocomputing*, vol. 51, pp. 161–180, 2003.

[19] T. M. Lehmann, C. Gönner, and K. Spitzer, "Survey: interpolation methods in medical image processing.," *IEEE Trans. Med. Imaging*, vol. 18, no. 11, pp. 1049–1075, 1999.

[20] P. Thevenaz, T. Blu, and M. Unser, "Interpolation revisited [medical images application]," *IEEE Trans. Med. Imaging*, vol. 19, no. 7, pp. 739–758, 2000.

[21] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust.*, vol. 29, no. 6, pp. 1153–1160, 1981.

[22] J. W. Hwang and H. S. Lee, "Adaptive image interpolation based on local gradient features," *IEEE Signal Process. Lett.*, vol. 11, no. 3, pp. 359–362, 2004.

[23] R. G. Casey, "Moment Normalization of Handprinted Characters," *IBM J. Res. Dev.*, vol. 14, no. 5, pp. 548–557, 1970.

[24] S. B. Park, J. W. Lee, and S. K. Kim, "Content-based image classification using a neural network," *Pattern Recognit. Lett.*, vol. 25, no. 3, pp. 287–300, 2004.

[25] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. Huang, "Large-scale image classification: Fast feature extraction and SVM training," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1689–1696.

[26] P. Angelov and R. Yager, "A new type of simplified fuzzy rule-based system," *Int. J. Gen. Syst.*, vol. 41, no. 2, pp. 163–185, 2011.

[27] P. P. Angelov, X. Gu, J. Principe, and D. Kangin, "Empirical data analysis - a new tool for data analytics," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2016, pp. 53–59.

[28] X. Gu, P. P. Angelov, A. M. Ali, W. A. Gruver, and G. Gaydadjiev, "Online evolving fuzzy rule-based prediction model for high frequency trading financial data stream," in *IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, 2016, pp. 169–175.

[29] P. Angelov and X. Zhou, "Evolving fuzzy-rule based classifiers from data streams," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 6, pp. 1462–1474, 2008.