ISSN 2007-9737

Exploring Convolutional Neural Networks Architectures for the Classification of Hand-Drawn Shapes in Learning Therapy Applications

Dionisio Ruíz Vazquez, Graciela Ramírez Alonso, Luis Carlos González Gurrola, Raymundo Cornejo García, Fernando Martínez Reyes

Universidad Autónoma de Chihuahua,
Facultad de Ingeniería,
Mexico

{galonso, lcgonzalez, rcornejo, fmartine}@uach.mx,
dio.ruiz11@gmail.com

Abstract. A positive consequence of the existence of a more inclusive society is the appearance of protocols to help identify those in need. One of these protocols is the application of tests with the aim to detect learning disabilities in children so that opportune intervention could be made. These advances, though, also pose challenges to the responsible to administer these tests, for example, the large number of tests to evaluate that make the process lengthy at the best. In this study, we implement a Convolutional Neural Network (CNN) model for the automatic classification of hand-drawn drawings of the Bender Gestalt Test (BGT), which is a test that evaluates the perceptual-motor maturity and perceptual disorder on individuals. In BGT, nine different drawings are presented to the patient who must reproduce them using pencil and paper. This study focuses on the automatic detection of the traces and their classification, then aiming to expedite the test evaluation process. Our proposed task-specialized CNN, named CNN4-Bender, is compared against other eleven neural-network based models registering an average performance of 91.56%, surpassed only by ResNet50 but with a high computational cost in this last one. To further evaluate our model we also consider other classification tasks that include the MNIST and OIHACDB datasets, where the CNN4-Bender architecture obtains a competitive performance and in some cases outperforms state-of-the-art models.

Keywords. Bender dataset, convolutional neural networks, automatic classification of Bender drawings.

1 Introduction

Hand-drawn images is a very complex neuromotor process that involves both cognitive and motor skills. Psychological and Neurological specialists have developed different neuromotor tests to detect specific parts of the brain that are working abnormally and identify diverse impairments in patients. In most cases, these problems are related to children exhibiting developmental deficits or adults with degenerative brain illness [10]. The Bender Gestalt Test, BGT, is one particular hand-drawn examination where the patients are asked to copy nine different geometric patterns [11].

When the patient finishes, the specialist identify if all the drawings were completed, then, he/she classify and interpret them by analyzing some properties (i.e. rotation, distortion, and angulation) to assign a score to those drawings. Commonly, all these previously described activities are manually performed and its completion is a time-consuming process [33].

The availability of an automated system able to classify and/or score the BGT examination could reduce significantly the workload of psychologists (i.e. segmenting and classifying these drawings), thus enabling the specialists to focus on the personal interaction with the patient. At the same time, the development of a BGT with

Automated Classification and Evaluation opens up the possibility to be applied by the patient's relatives at the peace of home, allowing them to closely monitor the progression of the patient's visuomotor skills.

The process to automatically classify hand-drawn drawings based on digitized images is known in the literature as an Optical Character Recognition (OCR) system [6]. These systems recognize different patterns based on specific features of the images. This is a complex process, but fortunately, it is a field with a wide support from the research community with contributions in English, French, Japanese, Chinese, Persian, Indic and Arabic languages, to mention some [3, 6, 21, 30, 38].

Recently, in order to evaluate novel recognition systems, some researchers have proposed diverse datasets, for instance, MNIST[1], PHOND [38], USPS[2], CVL Single Digit dataset of ICDAR-2013 [14], HACDB [24], and CASIA-HWDB/ OLHWDB [30] are some good examples that have even become a standard benchmark for the community. In this study, we introduce another dataset that besides being challenging also presents an important application, this being the BGT dataset.

Since classifying the BGT traces could be considered a novel computational task, we are the first to propose a specific model architecture for this problem, which is based on a Convolutional Neural Network (CNN). CNN's have proved to offer very competitive results for similar tasks since they try to cope with issues related to shifting, scale, and distortion that may affect the performance of image classifications algorithms [25].

Based on a scanned image of the BGT drawings, our implementation identifies their location and the CNN classify them considering noisy contexts including rotations, scaling, translation, changes of stroke direction, curvature and length of the lines. To evaluate our model we compare it against another eleven neural-network based models. Results suggest that the CNN is a robust and effective tool to be used by specialists on the BGT. The major contributions of this study are:

- We introduce the Bender Gestalt Test as a classification problem considering it as an original application in the context of learning therapy application that could represent a meaningful task for the community.
- We present practical baseline results on this problem using Convolutional Neural Networks, which could dictate further advances in this direction.
- We make publicly available the BGT dataset. We expect that given the importance of solving this task, the dataset could be eventually integrated as another benchmark for character recognition tasks.

The scope of this paper is only in the classification of the nine different hand-drawn images of the BGT test in order to compare the performance of our proposed model with state-of-the-art neural architectures that perform similar tasks. The rest of this paper is organized as follows.

Section 2 reviews literature regarding different machine learning approaches that have been applied in optical character recognition (OCR) systems. Section 3 describes the Bender Gestalt dataset and the system developed to automatically detect the Bender drawings in the scanned paper. Section 4 explains the implementation of the different computational models used in this work. Section 5 presents the details of our experimentation, and Section 6 offers concluding remarks.

2 Related Work

The different techniques used to recognize a handwritten character could be categorized into two streams: feature extraction or classifier approaches. In feature extraction approaches, the main contribution is in the methodology used to learn robust features. In classifier approaches, the objective is to propose an improvement in the machine learning model. In this section, will be described some important contributions implemented in OCR systems considering this categorization.

[1] http://yann.lecun.com/exdb/mnist/

[2] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.htmlusps

2.1 Models based on Robust Features Extraction Approaches

The paper presented in [38] by Sajedi describes a methodology where 96 different statistical and structural features are extracted for each digit of the PHOND database. Statistical distribution of pixels, zoning, moments, direction histograms, direction of strokes, endpoints, and intersection of loops are considered as features. Different configurations of the KNN and Support Vector Machine (SVM) classifiers were implemented obtaining an average recognition rate of 97.89%. Cheng-Lin et al. [31] use three different datasets, combine ten feature vectors and eight classifiers to recognize handwritten characters.

The best results reported were obtained with the chaincode and gradient features algorithms in combination with the SVM classifier. The chain code histogram feature algorithm was also implemented by Pramanik and Bag in [34]. The authors reported that a Multi-Layer Perceptron based classifier achieved a performance of 88.74% in the recognition of Bangla characters. Inkeaw et al. [20] implemented the Least Absolute Shrinkage and Selection algorithm in combination with the Histogram of Oriented Gradients to form a feature vector. The Local Preserving Projection algorithm was used to reduce the dimensionality of the feature vector that is the input of a SVM algorithm. The authors use the Lanna Dhamma and Thai datasets achieving an accuracy of 82.49% and 70.74% respectively.

2.2 Models based on Classifier Approaches

A *K*-nearest neighbor (KNN) graph based on the Image Distortion Model Distance (IDMM) algorithm is presented by Cecotti in [5] to classify digit images. By using only 332 labeled images of the MNIST database he obtained an accuracy of 98.54% on the labeling of the remaining training dataset and an accuracy of 99.10% on the test dataset. In [18] the authors proposed an algorithm able to perform well when trained and tested on data from different datasets such as MNIST, USPS, and CVL. The classifier was based on the SVM algorithm with an automatically adaptation on its parameters.

The mean recognition rate reported was 89.86%. Aspiras and Asari presented in [2] a new neural network architecture, the hierarchical auto-associative polynomial neural network (HAP Net), which provides a nonlinear weighting. The authors validate their proposal with the MNIST database surpassing state of the art models.

2.2.1 Models based on Deep Learning Techniques

The correct classification of characters in OCR systems depends highly on the feature extraction module as demonstrated in [9]. The success of implementations based on CNN is attributed to the multi-scale high-level image representation obtained in the convolutional layers. Therefore, a CNN learns discriminative representations from raw data eliminating the need to define specific hand-craft features.

This is the case of Elleuch et al. [15] that presented a model based on CNN and SVM for offline Arabic handwriting recognition. The model was validated with the HACDB and IFN/ENIT databases achieving an error of 6.59% and 7.05%, respectively. In [39] is proposed a multi-column multi-scale CNN for the recognition of isolated characters and digits of Indic scripts. Their method was evaluated on nine different datasets achieving superior recognition rates when compared with state-of-the-art methods.

Xiao et al. [41] implemented a CNN for the recognition of Chinese characters. They proposed a global supervised low-rank expansion method and an adaptive drop-weight to reduce the computational cost and compress the network. Their model was 30 times faster and 10 times more cost efficient compared with state-of-the-art CNN with only a 0.21% drop in accuracy.

Similarly, Yang et al. [43] implement a CNN to recognize Chinese characters with a drop sample training, five convolutional layers, and two fully connected layers. Xu-Yao et al. [44] implemented a strategy based on the directMap algorithm and a CNN by using the HCCR database of ICDAR 2013. The result reported achieve the best results and surpass human-level performance.

Lin et al. [27] propose a deep model CAPTCHAs recognition structure to learn Chinese characters. The authors demonstrate that the CNN approach can handle difficult distortion issues. Qu et al. [36] integrate a new combination of directional feature maps with a CNN of 9 layers by using the IAHCCR dataset. The authors in [32] implemented a deep network with residual connections to identify Chinese Characters (ICDAR 2013 offline HCCR dataset). A new data generation strategy was proposed where the characters are recombined to form new ones.

The authors reported an accuracy of 97.53%. Boufenar, Kerboua, and Batouche [4] implemented a CNN to identify Arabic characters by using an expanded version of the HACDB dataset called OIHACDB-40 and the AHCD dataset. They implemented three different CNN strategies outperforming all other state-of-the-art methods. Kavitha and Srimathi implement a CNN to recognize handwritten Tamil characters achieving a recognition rate of 95.16% [22].

Based on these results, in this paper, we implemented different CNN architectures to classify the nine drawings of the BGT. This is particularly important due to the nature of our dataset. Drawings of patients with learning disabilities could vary widely since patients would present different writing styles and learning disabilities; therefore it is difficult to define an appropriate selection of hand-craft features. With the use of CNN, these features could be automatically extracted from the images making them ideal for this type of problems.

3 Bender Gestalt Test

The BGT, published by Dr. Bender in the 1930s, evaluate the perceptual-motor, visuospatial, visuoconstructive, visual-motor maturity and perceptual distortions associated with various neurological disorders [1, 10, 11]. It consists of nine cards, each presented separately to the child who is asked to copy the designs, one at a time, on a sheet of paper. Fig. 1 shows these BGT cards. The paper on which each of the nine designs are copied records a perceptual experience of the child spontaneous movement patterns, the primitive motor processes and developing skills, with it's correspondent constructive energies [11].

3.1 Bender Gestalt Dataset

The data were collected from 323 different children whose age varies between 5 and 11 years. 86 children attend language and learning therapies in a center located in Northern Mexico whereas the rest of the data were collected from an inclusive elementary school. The paper where the patients performed the BGT was scanned at a resolution of 300 dpi. The initial dataset then is built using the raw drawings. That is, each piece of paper shows the individual's hand drawing competence, space organization, notions of scale and orientation. The scope of this work is to implement the system in the therapy of the patient in an automated manner. Most researchers segment and cropped the digits of their databases manually but we prefer to perform this step with minimal user intervention.

3.1.1 Detection of Bender Drawings

A preliminary step towards the implementation of a classification system is the development of a preprocessing phase consisting of the detection of drawings. For this task, the Prewitt algorithm [35] is implemented to detect the edges in the image. The scanned RGB image is transformed to grayscale $\mathbf{I}_g(x,y)$ and convolved with the Prewitt edge detection masks. Eq 1 shows this operation for the horizontal and vertical directions:

$$\begin{aligned}\mathbf{G}_r(x,y) &= \frac{1}{3}\begin{bmatrix}-1 & 0 & 1\\ -1 & 0 & 1\\ -1 & 0 & 1\end{bmatrix} * \mathbf{I}_g(x,y),\\ \mathbf{G}_c(x,y) &= \frac{1}{3}\begin{bmatrix}-1 & -1 & -1\\ 0 & 0 & 0\\ 1 & 0 & 1\end{bmatrix} * \mathbf{I}_g(x,y).\end{aligned} \tag{1}$$

The magnitude of the Prewitt operator is defined as $\mathbf{G}(x,y) = \sqrt{\mathbf{G}_r^2 + \mathbf{G}_c^2}$. If this magnitude surpasses a threshold value, the pixel is classified as an edge $\mathbf{E}(x,y)$.

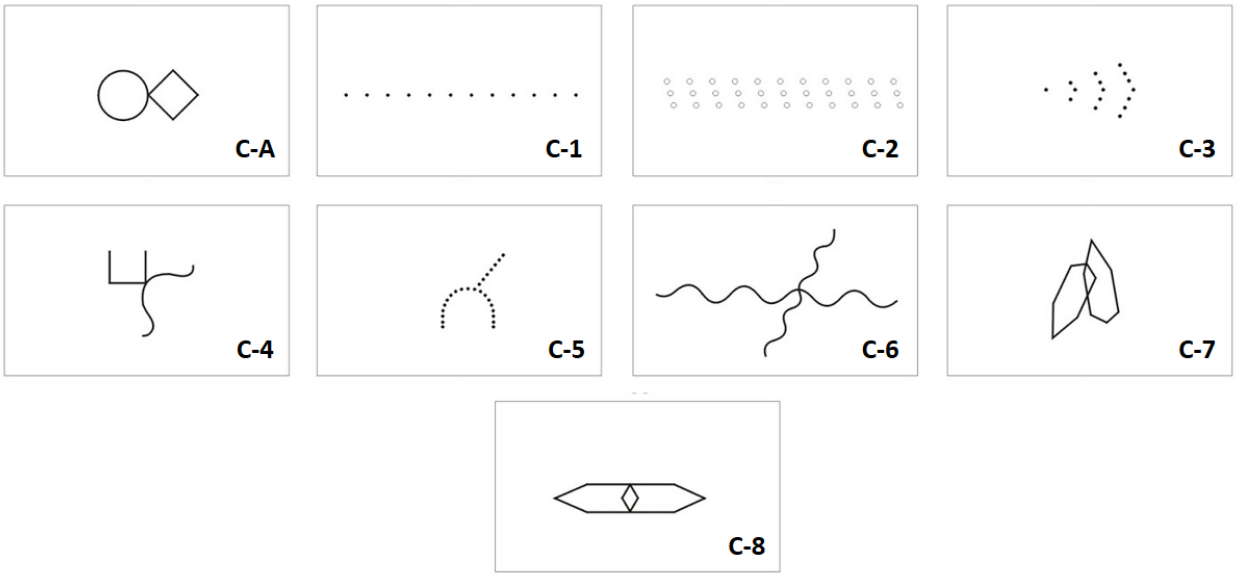

Fig. 1. Figures of the Bender Gestalt Test

Later, the morphological operator of dilation followed by the fill holes and erosion functions perform a more accurate detection of Bender drawings.

Let $\mathbf{S}(m,n)$ define the morphological structure element of size 59×59, representing a disk-shaped mask with a radius of 30. The binary dilation operator of **E(x,y)** by $\mathbf{S}(m,n)$ is defined by:

$$\mathbf{E} \oplus \mathbf{S} = \{z \mid [(\hat{\mathbf{S}})_z \cap \mathbf{E}] \neq \emptyset\}. \tag{2}$$

The dilation of **E** by **S** is the set of all displacements, z, such that the reflection of **S** (or $\hat{\mathbf{S}}$) and **E** overlap by at least one element. The erosion operator $\ominus$ reduces the size of objects and is defined as:

$$\mathbf{E} \ominus \mathbf{S} = \{z \mid (\mathbf{S})_z \subseteq \mathbf{E}\}. \tag{3}$$

The erosion indicates the set of points z such that **S**, translated by z, is contained in **E**. Finally, an analysis of areas finds those regions that have a high probability to be one of the nine draws. This analysis is as follows, the area of the largest object is identified and the area of the rest of the regions are divided by it. Only those regions whose result surpass the value of 0.1 are considered as a possible section that may contain a BGT drawing. Fig. 2 shows a block diagram of the pre-processing stage.

There are some cases where the automatic detection of Bender drawings is prone to error. Circumstances such as the quality of the paper, brightness, or factors as if it is a recycled piece of paper or situations where the drawings are very close to each other, cause an inaccurate segmentation of drawings. Fig. 3 shows an example of this situation. In this case, a shadow region produced by the scanner causes the erroneous detection of two regions (bottom of the paper). The effect of this shadow and the location of drawings (too close) caused that all of them were enclosed within the same rectangle.

A semi-automatic strategy was implemented in order to cope with the above issues. Each time the system encloses a region, the user must confirm that this selection correctly encloses the drawing. If this is not the case, the system will ask the user to select the correct region. Also, at the end of the process, the system asks the user if all the drawings were already identified. If any drawing is missing, the user will define the number of absent traces and they must be selected

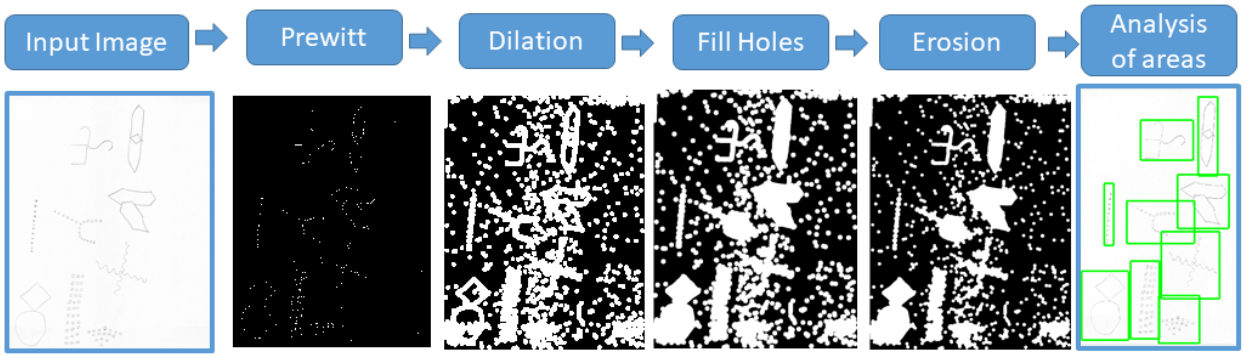

Fig. 2. Block diagram of the automatic detection of Bender drawings.

manually. With this strategy, we guarantee the correct identification of all the Bender drawings.

4 Network Architecture Selection

Some of the networks used in the field of image processing to recognize numbers, digits or are LeNet [25], AlexNet [23] and VGGNet [40], to mention some. In order to identify the most appropriate CNN architecture to our problem, we performed an experiment with five different architectures adapted to our BGT dataset. A Multilayer Perceptron (MLP), a simple CNN, a CNN based on the LeNet-5 network with slight modifications, a CNN based on the AlexNet architecture and our task-specialized CNN proposal named CNN4Bender.

CNN4Bender was developed as a neural model with fewer parameters than AlexNet but with the ability to achieve a good performance in the classification task. Also, CNN4Bender includes a dropout layer as a regularization technique for reducing overfitting and Batch Normalization layers. The notation used to represent the CNN will be explained next: *K*@*d*-***Cs*** represents a convolutional layer with K kernels with dimensionality @*d*, and a stride of s pixels.

M**P**s denotes a max-pooling layer with a $M \times M$ pooling window and a stride of s pixels. ReLU denotes the Rectified Linear Unit Activation layer, BN refers to a Batch Normalization Layer and n**FC** denotes a Fully Connected layer with n neurons. The description of the different CNNs is presented in Table 1.

4.1 Data Augmentation

Once the nine drawings of the scanned page are detected and separated, they become the input to the Convolutional Neural Network. In total, we have 2,850 Bender drawing patterns. An important factor to consider to train a CNN is the size of the dataset. It is difficult to define an appropriate size, for example, the MNIST database has 70,000 images of ten different digits, 60,000 to train and 10,000 to test. The Bangla CMATERdb 3.1.3.3 database has 117 classes, 34,439 training samples and 8,520 test samples [37]. The HACDB database has 6,600 shapes, 5,280 to train and 1,320 to test [24]. The OIHACDB-40 database has 30,000 images evenly divided into 40 classes [4]. Considering this, we implemented a data augmentation strategy to increase our dataset. Data augmentation methods generate a large number of training data using label-preserving transformations, such as cropping and flipping.

By implementing data augmentation, we can reduce overfitting and improve the generalization ability of the CNN [16]. The data augmentation technique implemented in our system consists of elastic transformation, local distortion, and random affine. Table 2 shows the number of samples that we have in our BGT database for each drawing before and after data augmentation. The first column shows the drawing identification, the second column the number of original samples,

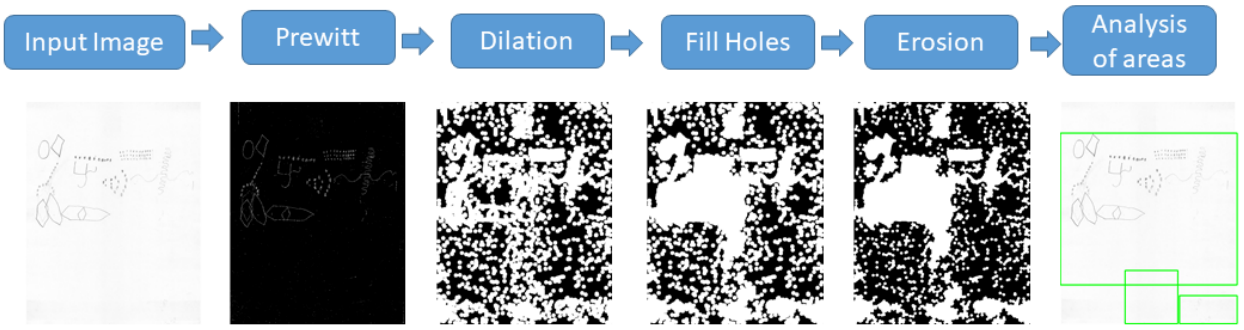

Fig. 3. Example of an erroneous detection of hand-drawn shapes

Table 1. Neural architectures used in our experiments

Neural Network	Architecture
MLP	512**FC** → ReLU → 512**FC** → ReLU → 9**FC** → SoftMax
CNN1	20@*5-**C1*** → RELU → 2**P**2 → 9**FC** → SoftMax
LeNet-Bender	6@*5-**C1*** → RELU→2**P**2 → 16@*5-**C1*** → RELU → 2**P**2 → 120@*1-**C1*** → RELU → 84**FC** → RELU → 9**FC** → SoftMax
AlexNet-Bender	96@*11-**C1*** → RELU → 2**P**2 → 256@*5-**C1*** → RELU → 2**P**2 → 384@*3-**C1*** → RELU → 384@*3-**C1*** → RELU → 4096**FC** → RELU → 4096**FC** → RELU → 9**FC** → SoftMax
CNN4Bender	48@*11-**C1*** → RELU → 2**P**2 → BN → 128@*7-**C1*** → ReLU, → 2**P**2 → BN → 256@*3-**C1*** → ReLU → BN → 1024**FC** → ReLU → *Dropout 0.5* → 1024**FC** → ReLU → 9**FC** → SoftMax

and the third column the number of samples after data augmentation. In total, we have 28,500 samples, an average of 3,166 per class. Considering that the HACDB has 750 images per class, and the Bangla CMATERdb has an average of 294 per class, we estimate that 3,166 is an acceptable quantity of samples. The Bender dataset can be found at this site[3].

Table 2. Number of samples for each class in our Bender dataset

Drawing	# Original samples	# Augmented samples
C-A	320	3200
C-1	318	3180
C-2	318	3180
C-3	314	3140
C-4	315	3150
C-5	316	3160
C-6	316	3160
C-7	315	3150
C-8	318	3180
Total	2,850	28,500

5 Experimental Results

All the different CNN architectures utilized in this work were implemented in Keras, a high-level neural network API [8]. The system ran on a Dell XPS 8920, Intel i7 7700, NVIDIA GeForce GTX 1080, 32GB RAM, 1TB ssd NVMe. The datasets used in the experiments are MNIST, OIHACDB-40, and our BGT. MNIST is a very popular dataset to evaluate different CNN architectures. The OIHACDB-40 dataset contains traces very similar to those of BGT. For the OIHACDB-40 and the created BGT datasets, a k-fold cross-validation

[3] https://github.com/GracielaRamirezA/Bender-dataset

with a shuffle strategy was used to separate the training and test datasets.

80% of the data was used for training and 20% for testing. In our implementation, we considered important that the testing data only include original BGT traces. The method of mini-batch stochastic gradient descent with a categorical cross-entropy objective function was implemented in our experiments.

The mini-batch size was set to 128, the learning rate was initiated at 0.01 and the maximum number of epochs was defined as 40.

5.1 Experiment 1: CNN Model Selection based on Random Initialization of Weights

In this experiment, the weights of the different neural models were randomly initialized and a MLP network was implemented as a baseline model. The architecture of the MLP presented in Table 1 obtained the best performance from different combination of neurons and activation functions in the hidden layers.

Table 3 presents the average results achieved with the $k = 5$ testing folds. CNN4-Bender architecture obtained the best performance in $accuracy$ and $f - measure$ with only 60,646,185 parameters.

The CNN1 architecture is the model with fewer parameters but with the lower performance. The LeNet-Bender and AlexNet-Bender models obtained a very similar performance, but the last one has more parameters.

By considering these results, the CNN4Bender architecture was selected as the best option to use in the following experimentation.

5.2 Experiment 2: Performance Comparison of the CNN4Bender Architecture using the OIHACDB-40 and MNIST Datasets

Part of the neural nets presented in [4] use the OIHACDB-40 dataset that contains 30,000 images of Arabic characters evenly divided in 40 classes (750 images for each class). The recognition of Arabic Handwritten characters is difficult due that it contains 28 letters with no upper and lower case and each letter has two or four different shapes. Additionally, 15 of the 28 letters have one or more points. The experiment that we performed consisted of using the CNN4Bender architecture and train it with the OIHACDB-40 and MNIST datasets. In both cases, the weights of the networks were randomly initialized and then trained with the corresponding data. Table 4 presents the results obtained.

Table 5 presents the results of different CNN architectures that used the MNIST database to validate their model proposals. Recent publications validate their models with different datasets in order to prove their robustness. Even when the scope of this work is not to propose a CNN model that surpass state of the art models, we found that our model is very competitive and in some cases surpass recent proposals.

Xu et al. [42] presented a SparseConnect idea to alleviate the over-fitting issue by sparsifying connections in the FC layers. The average performance that they reported with the SparseConnect1 strategy and MNIST database is 98.82% (best performance reported is 99.56%); with SparseConnect2 the average performance is 96.46% (best performance is 99.77%). Lee, Park and Sim [26] proposed a method to tune the hyper-parameters of the feature extraction module of a CNN by implementing a parameter-setting free harmony search algorithm. The results reported with the MNIST database achieved an accuracy of 99.25 %. Chevalier et al. [7] presented the DeepLUPI where the loss function is based on multi-class coefficients measures that depend on the difficulty to correctly recognize an input image. The best performance reported by the authors is 99.05% (95 errors).

As can be observed, the performance achieved with our CNN4Bender model surpass two of these new proposals where different databases are used to validate them. Regarding the results with OIHACDB, the CNN4Bender model achieved a performance of 98.33%, the authors in [4] reported an accuracy of 100 % but they only use this database to validate their models.

Table 3. Average results by considering random initialization of weights and $k = 5$ testing folds

CNN	Parameters	Image Size	Accuracy	F-measure
MLP	4,677,641	90×90	81.81%	81.76%
CNN1	333,429	90×90	78.69%	78.72%
LeNet-Bender	3,644,429	90×90	89.11%	89.10%
AlexNet-Bender	171,543,625	90×90	89.81%	89.80%
CNN4Bender	60,646,185	90×90	91.57%	91.56%

Table 4. Performance of the CNNs trained with a random initialization of weights

CNN	Database	Image size	Weight init.	Accuracy
CNN4Bender	MNIST	90×90	random	99.39 %
CNN4Bender	OIHACDB	90×90	random	98.3332%

5.3 Experiment 3: Transfer Learning, Use of the Weights of the Trained Networks as Feature Extractors

Taking as an assumption that a trained model has already learned some features, we can take advantage of this in our problem. Specifically, the first transfer learning strategy implemented in this work is based on using the convolutional layers of a trained CNN as a feature extractor. The network will be trained only on the Fully Connected layers by using a new dataset. In CNN architectures, the first convolutional layers extract global features, whereas the final layers of the network detect specific details of the images. In this part of the experimentation, we considered the weights previously trained with the MNIST and OIHACDB-40 databases and include the samples of the Bender dataset to train the FC layers.

5.4 Experiment 4: Transfer Learning, Use of the Weights of the Trained Networks for the Fine-tuning Strategy

Another common approach used in transfer learning schemes is to fine-tune the weights of a trained network with a low learning rate. By considering a new dataset, the network will slightly modify the previously learned weights to adapt it to the new knowledge. By doing this, we avoid the problem to randomly initialize the weights of the network. As in the previous experiment, there were considered the samples of the Bender dataset.

Table 6 presents the results obtained with the two transfer learning strategies. The best results were obtained by applying a fine-tuning with the MNIST and Bender datasets.

5.5 Experiment 5: Comparisons with VGG, ResNet and DenseNet Models

It is common to find specific CNN architectures developed to solve problems related to image classification tasks. Models such as VGG [40], Residual Networks (ResNet) [17], and Dense Convolutional Networks (DenseNet) [19], are some of them.

VGG16 is one of the first deep CNN architectures. It consists of 16 convolutional layers and is commonly used in the computer vision community for extracting features. The main disadvantage is it's 138 million of parameters that makes it very slow to train.

ResNets are deeper models than VGG nets. A ResNet has shortcuts or skip connections that allows the gradient to be directly backpropagated to earlier layers helping the vanishing-gradient problem of deep models. ResNet has achieved impressive performance with different datasets, such as ImageNet [13] and COCO object detection [28]. There are two types of blocks used in ResNet: identity and convolutional blocks. ResNet50 is a Residual Network of 50 layers and it is used in our experimentation as a reference model.

In DenseNet, the connection is from one layer to all its subsequent layers through concatenation. This network has also been implemented to resolve the vanishing-gradient problem. A DenseNet has q dense blocks that consist of multiple convolution layers followed by transition

Table 5. Comparison of the CNN4Bender model against recent proposal considering the MNIST database

Database	CNN4Bender	Xu et al. [42]	Lee et al. [26]	Chevalier et al. [7]
MNIST	99.39 %	99.77%	99.25 %	99.05%

Table 6. Results obtained with the Transfer Learning strategies

Database	Image size	Weight init.	Accuracy
OIHACDB-40 and Bender	90×90	Transfer Learning (FC)	89.614%
MNIST and Bender	90×90	Transfer Learning (FC)	88.982%
OIHACDB-40 and Bender	90×90	Transfer Learning (FineTuning)	89.789%
MNIST and Bender	90×90	Transfer Learning (FineTuning)	90.701%

layers. The DenseNet model used in our experiments has three dense blocks with a grow-rate equal to 12. The input image is first convolved with 24 filters with a kernel size of 3x3, followed by a batch normalization layer, a RELU activation function and a max pooling operation with a stride = 2.

The transition layers have a batch normalization layer, a RELU activation function, 1x1 convolutional filters and the average pooling operation with a stride = 2. To perform the classification task, the last dense block has a global average pooling layer and a softmax activation function.

5.6 Statistical Test

We applied a non-parametric Friedman test with a significance level of α=0.05 [12, 29] to analyze the differences in the algorithms. The Friedman test compares the average rank of algorithms under the null-hypothesis that states that all the algorithms are equivalent and so their ranks should be equal.

If the null-hypothesis is rejected, the Nemenyi test compares all classifiers with each other. The performance of any two classifiers is significantly different if the critical distance between them differ by at least the critical difference CD.

Table 7 presents the ranking of the twelve evaluated methods.

There are situations where two methods obtained the same number of errors distributed in different classes causing a tie in their ranking. Eq. 4 considers this situation in the statistic's calculus [29]:

$$F_r = \frac{12\sum_{j=1}^{k} R_j^2 - 3N^2k(k+1)^2}{Nk(k+1) + \frac{\left(Nk - \sum_{i=1}^{N}\sum_{j=1}^{g_i} t_{i,j}^3\right)}{(k-1)}}, \quad (4)$$

where g_i = number of sets of tied ranks in the i^{th} group, $t_{i,j}$ = size of the j^{th} set of tied ranks in the i^{th} group. N indicates the number of iterations that were evaluated, k is the number of models and R_j is the sum of the ranks of each model.

In this case, $F_r > \chi^2_{crit}$, this null hypothesis is rejected. The critical distance for the Nemenyi test is calculated as follows:

$$\begin{aligned} CD &= q_\alpha\sqrt{\frac{k(k+1)}{6N}} \\ &= 3.2680\sqrt{\frac{12(12+1)}{6(5)}} \\ &= 3.2680\sqrt{5.2} \\ &= 7.4522. \end{aligned}$$

q_α is the critical value defined in [12] for α = 0.05. In Fig. 4, all methods are ordered based on their respective rank, the best method is located rightmost. The methods for which there is no evidence of statistical difference are joined by the horizontal line.

Based on this analysis, we observe that our proposal, CNN4-Bender, is within the best models. If we consider a more rigorous evaluation of the ranking performance, only ResNet50 surpass CNN4-Bender. However, one of the drawbacks of ResNet50 is the time it takes to train it. This makes

Table 7. Ranking results obtained in each fold

Fold	MLP	CNN1	AlexNet-Bender	LeNet-Bender	CNN4-Bender	CNN4-B MNIST-FC	CNN4-B MNIST-FT	CNN4-B OIHACDB-FC	CNN4 OIHACDB-FT	VGG16	ResNet50	DenseNet
1	11	12	9	4	2	8	5	6.5	6.5	3	1	10
2	10	12	8	5	2	4	3	9	7	6	1	11
3	11	12	9	7	3	10	4	5.5	5.5	8	2	1
4	12	11	8	4	3	9	2	6	7	5	1	10
5	11	12	9	10	1	8	5	7	6	4	2	3
Sum	55	59	43	30	11	39	19	34	32	26	7	35
Average	11	11.8	8.6	6	2.2	7.8	3.8	6.8	6.4	5.2	1.4	7

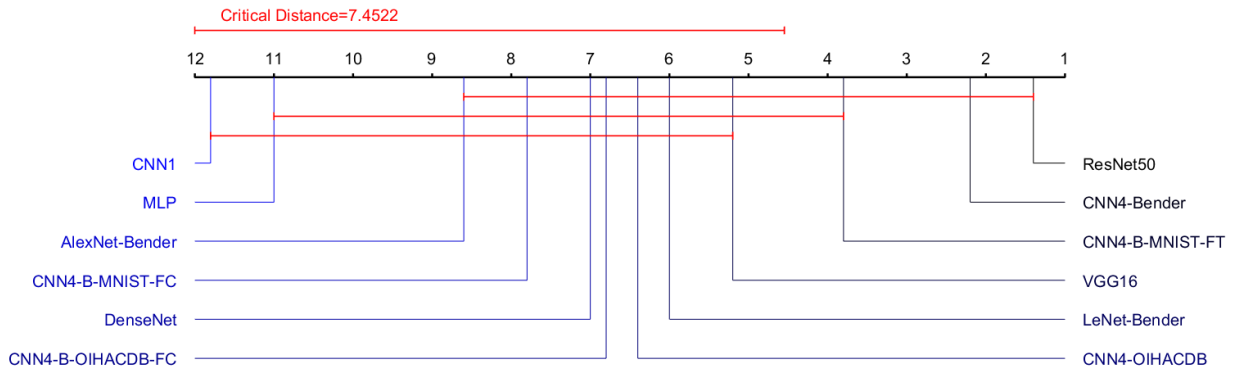

Fig. 4. Critical Difference diagrams for the comparison of the twelve models based on Friedman non parametric test with α=0.05

it impractical for applications where the hardware is limited.

For example, the training of one epoch took 20 s with the CNN4-Bender model, with ResNet50 it took 60 s. For the others state-of-the-art architectures, the training of 1 epoch took 65 s for VGG16 and 30 s for DenseNet.

Table 8 shows the confusion matrix of one of the folds of the CNN4-Bender model.

Table 8. Confusion matrix of the CNN4-Bender model considering one of the folds

	C-A	C-1	C-2	C-3	C-4	C-5	C-6	C-7	C-8	Acc. (%)
C-A	59	0	0	0	0	0	0	2	0	96.72
C-1	0	54	0	0	0	1	0	0	0	98.18
C-2	0	1	56	1	0	0	1	0	0	94.91
C-3	0	1	3	60	0	5	0	0	1	85.71
C-4	1	0	0	0	56	0	2	2	0	91.80
C-5	0	2	0	0	1	57	4	0	1	87.69
C-6	0	0	2	0	1	1	62	0	0	93.93
C-7	4	0	0	0	0	0	0	50	2	89.28
C-8	0	0	0	0	0	0	0	3	74	96.10
									Avg	92.70

Some errors of the CNN4-Bender model are shown in Figure 5. The first column shows a drawing of the C3 class assigned to C2. This drawing has more circles compared to the C3 Bender pattern, this could be a reason for this classification error.

The second column shows a C5 trace assigned to C6. One reason for this error could be the missing circles of the drawing caused a confusion with the C6 class. The C7 trace classified as C8 and C8 classified as C7 could be considered as “normal” because even for a human expert is difficult to classify correctly these traces.

6 Conclusions

The automatic recognition of hand-drawn images is a field widely explored in the literature where most of the computational efforts are oriented to the recognition of English, French, Japanese, Chinese or Arabic characters. Most of these strategies could be adapted to analyze the drawings of individuals in order to detect problems

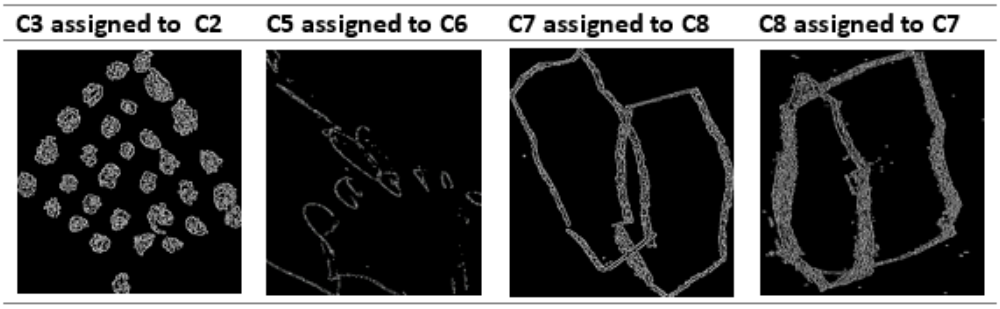

Fig. 5. Classification errors obtained with the CNN4-Bender model

related to intellectual disabilities, attention deficit or hyperactivity disorders. In this paper is presented a first attempt to detect and classify automatically the nine drawings of the Bender Gestalt Test based on computer vision algorithms. This is an important and original application of computational models. To the best of our knowledge, this is the first time that this problem is analyzed from a computational point of view by using sophisticated models such as Convolutional Neural Networks.

Our system initially identifies in the piece of paper the nine drawings of the BGT by performing an analysis of areas based on the scanned image. In this step, an edge detector and morphological algorithms are implemented. Some problems exist when the patient draws the traces too close. For this reason, we proposed a semi-automatic procedure where the user manually selects the location of traces.

When implementing CNN models, some strategies must be considered in order to increase the performance of the network. In this paper, there were implemented three of them: a data augmentation approach, transfer learning schemes and the use of Batch Normalization Layers. Data augmentation was implemented to increase the Bender database and avoid over-fitting problems.

The two transfer learning strategies were considered to test if previous knowledge would improve the classification accuracy. For the case of the Batch Normalization Layers, they were included because they have proven to increase the stability of the network by normalizing the output of hidden layers. This also helps to speed up the training process.

The results obtained with the comparison of twelve different architectures in the non-parametric Friedman test indicate that the best models were those based on CNN with more than one convolutional layer.

On the other hand, the architecture implemented in CNN4-Bender achieved the second best performance. ResNet50 obtained the best result, but it's training is three times slower than CNN4-Bender. Because there is not a significant difference between ResNet50 and CNN4-Bender, we conclude that CNN4-Bender is a better option to use in this particular classification problem.

The findings reported in this paper are very encouraging to be the first computational approach to classify automatically the nine drawings of the Bender Gestalt Test.

Not only simple CNN architectures were considered in the comparisons, recent deep architectures such as VGG16, ResNet and DenseNet models were also included. At the same time, the proposed CNN4-Bender model was evaluated with two other databases, MNIST and OIHACDB-40. The results demonstrate that this particular architecture achieved a good generalization with other datasets.

Acknowledgments

We would like to thank the support of the Jose David Institute, A.C. for providing the Bender Gestalt Test of their patients.

References

1. **Allen, R. A. & Decker, S. L. (2008).** Utility of the Bender visual-motor gestalt test—second edition in the assessment of attention-deficit/hyperactivity disorder'. *Perceptual and motor skills*, Vol. 107, No. 3, pp. 663–677.
2. **Aspiras, T. H. & Asari, V. K. (2017).** Hierarchical autoassociative polynimial network (hap net) for pattern recognition. *Neurocomputing*, Vol. 222, pp. 1–10.
3. **BAG, S. & HARIT, G. (2013).** A survey on optical character recognition for Bangla and Devanagari scripts. *Sadhana*, Vol. 38, No. 1, pp. 133–168.
4. **Boufenar, C., Kerboua, A., & Batouche, M. (2017).** Investigation on deep learning for off-line handwritten Arabic character recognition. *Cognitive Systems Research*.
5. **Cecotti, H. (2016).** Active graph based semi-supervised learning using image matching: Application to handwritten digit recognition. *Pattern Recognition Letters*, Vol. 73, pp. 76–82.
6. **Chaudhuri, A., Mandaviya, K., Badelia, P., & K Ghosh, S. (2017).** *Optical Character Recognition Systems for Different Languages with Soft Computing*, volume 352 of *Studies in Fuzziness and Soft Computing*. Springer International Publishing, Cham.
7. **Chevalier, M., Thome, N., Henaff, G., & Cord, M. (2018).** Classifying low-resolution images by integrating privileged information in deep CNNs. *Pattern Recognition Letters*, Vol. 116, pp. 29–35.
8. **Chollet, F. et al. (2015).** Keras. `https://github.com/keras-team/keras`.
9. **Cilia, N. D., Stefano, C. D., Fontanella, F., & di Freca, A. S. (2019).** A ranking-based feature selection approach for handwritten character recognition. *Pattern Recognition Letters*, Vol. 121, pp. 77–86.
10. **Cobrinik, L. (1987).** A process analysis of Bender gestalt test performance in childhood emotional disorder: A single-case study. *Child psychiatry and human development*, Vol. 17, No. 4, pp. 242–256.
11. **Cobrinik, L. (1988).** The Bender gestalt test in childhood emotional disorder. *Psychiatric Quarterly*, Vol. 59, No. 3, pp. 235–243.
12. **Demšar, J. (2006).** Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, Vol. 7, No. Jan, pp. 1–30.
13. **Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009).** Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255.
14. **Diem, M., Fiel, S., Garz, A., Keglevic, M., Kleber, F., & Sablatnig, R. (2013).** Icdar 2013 competition on handwritten digit recognition (hdrc 2013). *2013 12th International Conference on Document Analysis and Recognition*, pp. 1422–1427.
15. **Elleuch, M., Maalej, R., & Kherallah, M. (2016).** A new design based-SVM of the CNN classifier architecture with dropout for offline Arabic handwritten recognition. *Procedia Computer Science*, Vol. 80, pp. 1712–1723.
16. **Han, D., Liu, Q., & Fan, W. (2018).** A new image classification method using CNN transfer learning and web data augmentation. *Expert Systems with Applications*, Vol. 95, pp. 43–56.
17. **He, K., Zhang, X., Ren, S., & Sun, J. (2016).** Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.
18. **Hosseinzadeh, H., Razzazi, F., & Kabir, E. (2016).** A weakly supervised large margin domain adaptation method for isolated handwritten digit recognition. *Journal of Visual Communication and Image Representation*, Vol. 38, pp. 307–315.
19. **Huang, G., Liu, Z., v. d. Maaten, L., & Weinberger, K. Q. (2017).** Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269.
20. **Inkeaw, P., Bootkrajang, J., Marukatat, S., Goncalves, T., & Chaijaruwanich, J. (2019).** Recognition of similar characters using gradient features of discriminative regions. *Expert Systems with Applications*, Vol. 134, pp. 120–137.
21. **Kaur, H. & Kumar, M. (2018).** A comprehensive survey on word recognition for non-indic and indic scripts. *Pattern Analysis and Applications*, Vol. 21, No. 4, pp. 897–929.
22. **Kavitha, B. & Srimathi, C. (2019).** Benchmarking on offline handwritten tamil character recognition using convolutional neural networks. *Journal of King Saud University - Computer and Information Sciences*.

23. **Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012).** Imagenet classification with deep convolutional neural networks. *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, Curran Associates Inc., USA, pp. 1097–1105.

24. **Lawgali, A., Angelova, M., & Bouridane, A. (2013).** HACDB: Handwritten Arabic characters database for automatic character recognition. *European Workshop on Visual Information Processing (EUVIP)*, pp. 255–259.

25. **LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998).** Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2278–2324.

26. **Lee, W.-Y., Park, S.-M., & Sim, K.-B. (2018).** Optimal hyperparameter tuning of convolutional neural networks based on the parameter-setting-free harmony search algorithm. *Optik*, Vol. 172, pp. 359–367.

27. **Lin, D., Lin, F., Lv, Y., Cai, F., & Cao, D. (2018).** Chinese character CAPTCHA recognition and performance estimation via deep neural network. *Neurocomputing*, Vol. 288, pp. 11–19.

28. **Lin, T.-Y., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014).** Microsoft coco: Common objects in context. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8693 LNCS, Springer Verlag, pp. 740–755.

29. **Linebach, J. A., Tesch, B. P., Kovacsiss, L. M., et al. (2014).** *Nonparametric statistics for applied research*. Springer.

30. **Liu, C., Yin, F., Wang, Q., & Wang, D. (2011).** ICDAR 2011 Chinese handwriting recognition competition. *2011 International Conference on Document Analysis and Recognition*, pp. 1464–1469.

31. **Liu, C.-L., Nakashima, K., Sako, H., & Fujisawa, H. (2003).** Handwritten digit recognition: benchmarking of state-of-the-art techniques. *Pattern Recognition*, Vol. 36, No. 10, pp. 2271–2285.

32. **Luo, W. & Zhai, G. (2018).** Offline handwritten Chinese character recognition based on new training methodology. **Zhai, G., Zhou, J., & Yang, X.**, editors, *Digital TV and Wireless Multimedia Communication*, Springer Singapore, Singapore, pp. 235–244.

33. **Moetesum, M., Siddiqi, I., Masroor, U., & Djeddi, C. (2015).** Automated scoring of Bender gestalt test using image analysis techniques. *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, IEEE, pp. 666–670.

34. **Pramanik, R. & Bag, S. (2018).** Shape decomposition-based handwritten compound character recognition for Bangla OCR. *Journal of Visual Communication and Image Representation*, Vol. 50, pp. 123–134.

35. **Prewitt, J. M. S. (1970).** Object enhancement and extraction. In *Picture Processing and Psychopictorics*. New York: Academic Press, pp. 75–149.

36. **Qu, X., Wang, W., Lu, K., & Zhou, J. (2018).** Data augmentation and directional feature maps extraction for in-air handwritten Chinese character recognition based on convolutional neural network. *Pattern Recognition Letters*, Vol. 111, pp. 9–15.

37. **Roy, S., Das, N., Kundu, M., & Nasipuri, M. (2017).** Handwritten isolated Bangla compound character recognition: A new benchmark using a novel deep learning approach. *Pattern Recognition Letters*, Vol. 90, pp. 15–21.

38. **Sajedi, H. (2016).** Handwriting recognition of digits, signs, and numerical strings in persian. *Computers and Electrical Engineering*, Vol. 49, pp. 52–65.

39. **Sarkhel, R., Das, N., Das, A., Kundu, M., & Nasipuri, M. (2017).** A multi-scale deep quad tree based feature extraction method for the recognition of isolated handwritten characters of popular indic scripts. *Pattern Recognition*, Vol. 71, pp. 78–93.

40. **Simonyan, K. & Zisserman, A. (2014).** Very deep convolutional networks for large-scale image recognition. *CoRR*, Vol. abs/1409.1556.

41. **Xiao, X., Jin, L., Yang, Y., Yang, W., Sun, J., & Chang, T. (2017).** Building fast and compact convolutional neural networks for offline handwritten Chinese character recognition. *Pattern Recognition*, Vol. 72, pp. 72–81.

42. **Xu, Q., Zhang, M., Gu, Z., & Pan, G. (2018).** Overfitting remedy by sparsifying regularization on fully-connected layers of CNNs. *Neurocomputing*.

43. **Yang, W., Jin, L., Tao, D., Xie, Z., & Feng, Z. (2016).** DropSample: A new training method to enhance deep convolutional neural networks for large-scale unconstrained handwritten Chinese character recognition. *Pattern Recognition*, Vol. 58, pp. 190–203.

44. Zhang, X.-Y., Bengio, Y., & Liu, C.-L. (2017). Online and offline handwritten Chinese character recognition: A comprehensive study and new benchmark. *Pattern Recognition*, Vol. 61, pp. 348–360.

Article received on 13/04/2020; accepted on 14/09/2020.
Corresponding author is Graciela Ramirez-Alonso.