Automatic Recognition of Isolated Handwritten Digits by Hidden Markov Models

Djilali CHOGUEUR[1], Karima MAACHOU[1] and Aicha SAYAH[1]

✉ Chogueur.djilali@univ-adrar.edu.dz

[1]Mathematics and Computer Science Department, University of Adrar, Algeria

Received: 23 November 2020

Accepted: 22 December 2020

Published online: 30 December 2020

Abstract

Hidden Markov models (HMM) have proven to be effective in the field of speech recognition. Their application to offline handwriting recognition is a subject still open to research, the goal of which today is to decrease the error rate. HMMs are stochastic models that resist noise, variation, and the elasticity of shape, which is a fundamental requirement in handwriting. The general objective of our work is to apply the Hidden Markov model to the recognition of isolated handwritten digits. We present in this paper a system for recognizing isolated handwritten digits based on discrete HMMs of Left-Right and ergodic topology, by applying a vector quantization on the features extracted from the images of the digits by the K-means algorithm. Our approach is validated and evaluated on the MNIST corpus that we have used in its entirety. The experiments we have carried out have as arguments the topology of the HMMs, the number of states per model, the size of the Codebook, as well as the number of iterations.

Keywords Recognition, Isolated Handwritten Digits, Hidden Markov Model, Discrete HMM, Vector Quantization, Codebook.

1. Introduction

Researchers have been trying to bring the reading skill of a machine closer to that of humans for more than fifty years, and despite the considerable efforts expended throughout these years, the machine remains far behind man in the recognition of distorted or fragmented or misspelled handwriting. Optical character recognition (OCR) has undergone a revolution, and almost all of the constraints related to the way data is acquired, slow scanning, standardization of fonts and many others have been lifted in recent years. As a result, very high recognition rates have become achievable at high speed and at reasonable costs [1]. However, most OCR systems still cannot read degraded documents and handwritten words.

In addition to print character recognition systems, much research and investigation has been carried out around the world regarding handwriting recognition on-line and off-line. According to Schalkoff Robert [2] for offline handwriting recognition, a typical business application of such a system is the automatic reading of bank checks. The machine must be able to recognize and match amounts in numbers and letters. Another application such as automatic mail sorting machines for identifying postal codes in post offices. The current trend is to design machines whose competence in data acquisition interfaces is similar or almost similar to that of humans. This is how online handwriting recognition systems emerged, and they are of great commercial interest. These systems work with pen computers, tablets, cellphones and more, where data acquisition will be in writing and not via the keyboard or scanner.

A major factor in achieving high performance in automatic recognition of handwritten digits is feature extraction. According to Oliveira Luiz et. al. [3], the literature of handwritten digit recognition shows three categories of feature extraction: the first being the extraction of gray levels or the binary value of all the pixels in the image, used mainly by classifiers based on neural networks [4][5]. The second category is based on structural features and geometric shapes such as end points, intersections, loops, curves, measures of concavity and directional features [6][7][8]. The characteristics of the last class are the statistical characteristics, calculated by applying global transformations to an image. Among the typical mathematical transformations, moments [9][10], Fourier descriptors [11] and transformed into wavelets [12].

The classification step is as crucial as the extraction. Among the classification methods most successfully applied to the recognition of handwritten characters (number and letter), the model correspondence methods (calculation of similarity degree) [13][14], neural networks [15][16][17][18]. SVM (Support Vector Machines) [19][20], statistical methods like KNN (K nearest neighbors) [21], Bayesian networks [22], Markov hidden models (HMM) widely applied for speech recognition, and increasingly for the recognition of handwritten digits [23][24][25], methods based on reasoning of fuzzy sets [26], structural methods which include grammatical methods [27] and graphical methods [28], and finally the multi-classifier methods which combine several classifiers for the recognition of the handwritten character [29].

According to Chmielnicki and Stapor [30], the three main corpora of images used in handwritten character recognition are: CENPARMI [31], CEDAR [15] and MNIST [5]. They are widely used for recognition performance validation. The MNIST database is the most widely used for the evaluation of classification algorithms. We will also use it for the validation of our offline handwritten digit recognition algorithm. The best results in recognition rate recorded with MNIST are present as follows:

- Teow and Loe [32] achieved a recognition rate of 99.41%;
- Belongie et. al. [33] achieved a recognition rate of 99.37%;
- As for Dong et. al. [34], they had a recognition rate of 99.01%;

Our work aims to apply hidden Markov models to the automatic recognition of handwritten digits capable of recognizing the ten isolated handwritten digits, which we evaluate on the MNIST corpus. The success of the recorded Markov hidden models in speech recognition has led researchers to deploy it as an appropriate tool for handwriting recognition, as they are stochastic models that resist noise, and variations in speech. Also, the observation sequences can be of variable length, which is a fundamental requirement in handwriting.

2. MNIST Database

In the context of our work, the database used is that of offline handwritten figures MNIST available free of charge on the Internet [35]. It is made up of 70,000 images extracted from the National Institute of Standards and Technology database (NIST), split into two training databases made up of 60,000 images and test databases made up of 10,000 images (see table 1).

The characteristics of the MNIST database are: (see figure 1)

- Each figure is associated with an image in gray levels on 256 values;
- All images have the same size (28x28);
- Images are centered;
- The two bases, training and testing, were annotated by around 250 annotators.

Fig. 1 A sample of the MNIST database

Table 1 Distribution of individuals by class in MNIST

Class	Training	Test
0	5923	980
1	6742	1135
2	5958	1032
3	6131	1010
4	5842	982
5	5421	892
6	5918	958
7	6265	1028
8	5851	974
9	5949	974
Total	60000	10000

For better optimization, the training database has been split into two parts. One part comprising 40,000 images used for training the models and the other part comprising 20,000 images for validation. The test database is used, only, for the final evaluation of the models. The partition, thus, carried out of the MNIST database is illustrated in Table 2 below:

Table 2 Learning database (development and validation), test database, with the number of elements of the associated class

Class	Training Set 1 (Development)	Training Set 2 (Validation)	Test
0	3924	1999	980
1	4563	2179	1135
2	3943	2015	1032
3	4081	2050	1010
4	3909	1933	982
5	3604	1817	892
6	3975	1943	958
7	4125	2140	1028
8	3860	1991	974
9	4016	1933	1009
Total	40000	20000	10000

3. Our Proposed System

The main objective of the study is to implement an isolated handwritten digit recognition system based on hidden Markov models (HMM). The database (MNIST) used includes (see Table 2):

- 60,000 images for training divided into 40,000 images for the codebook generation and 20,000 images for validation.
- 10,000 test images to evaluate the model found.

First, a set of preprocessing operations is carried out in order to eliminate or at least reduce the noise in order to simplify the procedure of extraction (characteristics). Two topologies of DHMM (Discrete HMM) are used in our experiments, the ergodic HMMs and the Left Right HMMs. Each figure is modeled by an HMM.

Figure 2 below represents the overall functional diagram of our recognition system:

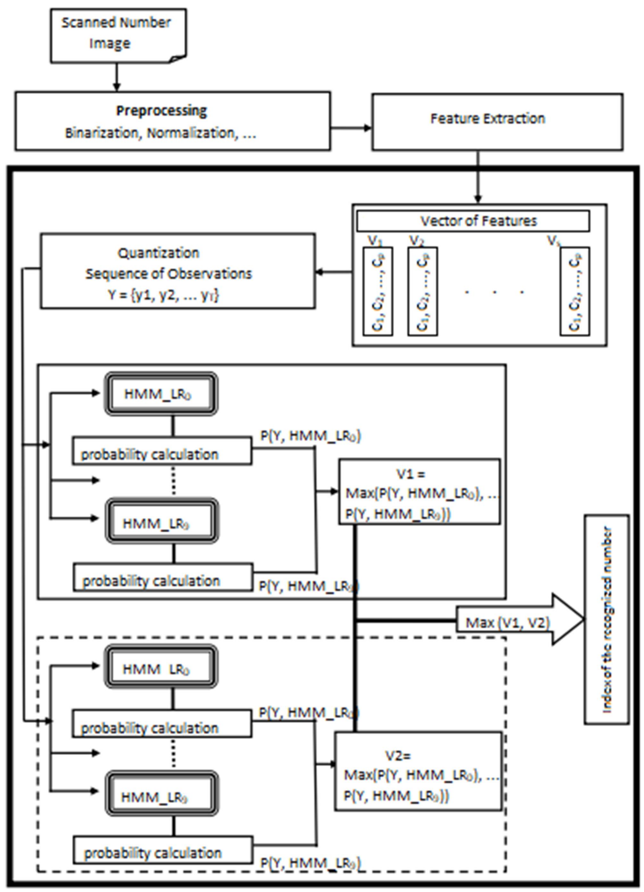

Fig. 2 A Architecture of the proposed system for recognizing isolated handwritten digits

3.1. Preprocessing

The preprocessings that have been applied to the raw data in our system are:

3.1.1. Binarization

Binarization is performed using the image processing toolbox from Matlab (see figure 3):

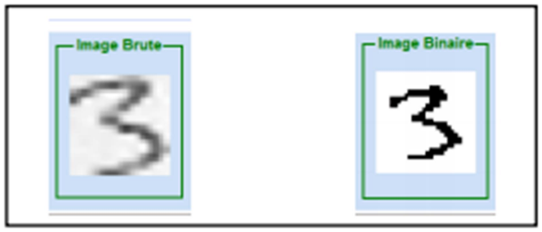

Fig. 3 Binarization of an image containing the number 3

3.1.2. Thinning

The figures acquired are of varying thickness, which influences the characteristics that distinguish them. This morphological operation is performed on an already binarized image. It consists of rendering all the digits one pixel thick (see figure 4):

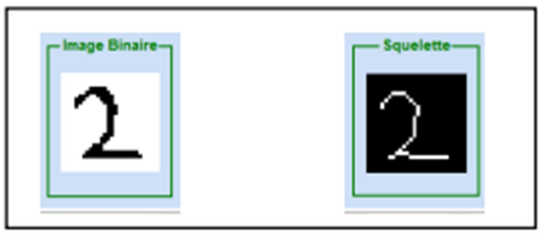

Fig. 4 Thinning of an image containing the number 2

3.1.3. Standardization

The size of the handwritten number varies from script to script. To make it uniform, we have applied size and centering normalization algorithms, (see figure 5):

Fig. 5 From raw image to normalization of number 6

3.2. Extraction of characteristics

The choice of the feature extraction method is an important factor in the realization of a recognition system. The method used in this study, was mainly inspired by the work of Blumenstein [36], devoted to feature extraction for isolated character recognition systems. The original image was zoned into 9 areas of the same size. Structural characteristics nine in number and geometric characteristics four in number are then extracted for each zone. The result of this step is a sequence of nine feature vectors of 13 components, each represented by a matrix of size (13, N * 9) where N is the number of images concerned by the extraction. Finally, the extracted characteristics are passed to the various HMMSs for training and recognition.

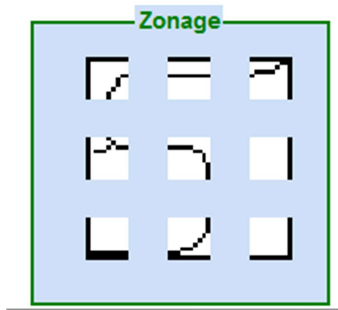

Fig. 6 Image of the number 5 divided into 9 zones

3.2.1. Extraction of structural characteristics

In order to provide a normalized input vector for the HMM classification system, the handwritten digit has been broken down into nine windows of equal size (zoning, see figure 6). For each zone, one extracts properties namely the vertical lines, the horizontal lines, the right diagonals and the left diagonals. In addition, the points of intersection between each type of line are located.

The line segment extraction algorithm first locates the starting point and intersection point in a particular window. It then proceeds to extract the number and length of the resulting row segments into an input vector containing nine floating point values. Each of the component values of the input vector (see figure 7) has been defined as follows:

1. The number of horizontal lines;
2. The normalized length of all horizontal lines;
3. The number of straight diagonal lines;
4. The normalized length of all straight diagonal lines;
5. The number of vertical lines;
6. The standard length of all vertical lines;
7. The number of left diagonal lines;
8. The normalized length of all left diagonal lines;
9. the normalized surface of the skeleton.

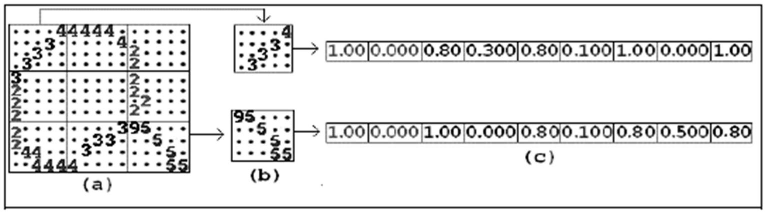

Fig. 7 (a) the processed image, (b) the zones, (c) vector of features (Blumenstein et al. 2003)

The number of rows of a particular type is normalized using the following method:

$$\text{Value} = 1 - 2(\text{number of rows} / 10) \tag{1}$$

The normalized length of a particular type of line is calculated as follows:

$$\text{Length} = \frac{\text{Total number of pixels in this line type}}{\text{Total number of pixels in the area}} \tag{2}$$

3.2.2. Geometric characteristics

After extracting the structural features, we move on to the geometric features by applying some functions based on regional properties using the region property functions of Matlab 7.0.1 namely:

- Regional Area: It is defined as the ratio between the number of pixels in the skeleton and the total number of pixels in the image.
- Eccentricity (Eccentricity): It is defined as the eccentricity of the smallest ellipse that fits the skeleton of the image.
- Orientation: Axis orientation, (see figure 8).
- The Euler number: It is defined as the difference between the number of objects and the number of holes in the image.

Fig. 8 The geometric property "Orientation"

3.3. Vector quantification and codebook generation

After the extraction step, taking the input feature vectors, the quantizer provides quantized observations as sequences to the discrete HMM decoder for recognition, using a codebook generated by the K- algorithm. means (K-means) which has four steps as follows:

3.3.1. Initialization step

Load the training set $\{v\ (n), 1 \leq n \leq M\}$ M feature vectors.

Put $m = 0$. Choose an initial set of centroid vectors $y_i(0), 1 \leq i \leq K$.

3.3.2. Classification step

Classify each of the training vectors $\{v\ (n), 1 \leq n \leq M\}$ M in the nearest cluster C_i.

$v\ \epsilon\ C_i(m), if\ d(v, y_i)\ \leq d\ (v, y_i), all\ j\ \neq\ i, 1 \leq j\ \leq K$

3.3.3. Code vector update step

$m\ =\ m\ +\ 1$. Update the centroid vector of each cluster by calculating its new centroid.

$y_i(m)\ =\ Cent\ (C_i(m)), 1 \leq\ i\ \leq L.$

3.3.4. Stop Test Step

If the decrease in the overall distortion (D_m) at iteration m relative to D_{m-1} is less than a certain stop threshold; otherwise go to the classification step. In the experiments we carried out three codebooks were created, using the training database made up of 40,000 images, or 9 * 40,000 = 360,000 vectors to be quantified. Given the limited memory used by Matlab 7.0.1 (max 2 GB), the maximum size of the codebook that we were able to generate is 232. The other two codebooks are 190 and 128 code vectors.

4. Recognition of the number

4.1. Learning

Learning makes it possible to adjust the different matrices and vectors to make them more faithful to a model, by passing the observations in an initial model and looking at how it behaves, seeing which states are privileged and which observations are made when. We have developed two types of HMM models: a left right type HMM and an ergodic type HMM.

Once the model topology has been constructed, it is necessary to choose the model parameters carefully because they influence the results. Since the size of the training corpus also influences the results, the training database chosen in this study is 20,000 images to have a suitable estimate of the model. The training algorithm is divided into three parts:

4.1.1. Initialization (coding)

This phase consists of sampling the input data by transforming the raw data into observations, which will be learned by the HMMs. For this, a zoning of 09 zones (size 3x3 pixels) is defined for each image. For each zone, the characteristics described in the section extraction of the characteristics are extracted. An image will be represented by nine feature vectors of 13 components each. Then the vectors resulting from the extraction of all the images, 9 * number of images, are quantified by assigning each one the index of the centroid which is closest to it in the appropriate Codebook. The result of this part is a set of discrete observation sequences.

4.1.2. Training

We can distinguish two different cases in the training problem depending on whether the structure (number of states and authorized transitions of the HMM) is known or not. If the structure is not defined, the training then becomes more difficult because it is necessary to deduce this structure from the examples provided to the training before setting it up. In the second case, the case of our study where the structure is known, the training is reduced to a training problem which consists in estimating the numerical parameters namely:

- The distribution of the probabilities of state transitions;
- The probability distribution of the observation symbols in each state;
- The distribution of the probability of being initially in a state.

The training of our models is carried out by the Baum-Welch algorithm based on Baum's theorem [37][38] which guarantees the achievement of a local optimum of the likelihood function by re-estimation of the parameters.

4.1.3. Registration

The parameters resulting from the previous part are saved in a database. This database will be used for recognition.

The diagram of the training system is shown in figure 9 below:

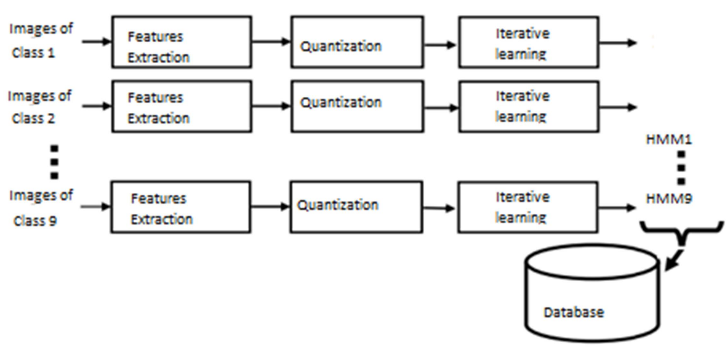

Fig. 9 Learning phase

4.1.4. Determination of the number of states:

After several experiments (34 experiments for each structure going from state 4 to state 37), we estimated the parameters of the two models Ergodic and Left-right, namely, the initial probabilities, the transition probabilities and the probabilities emission with a number of states N = 36 for the Left-Right HMM and N = 34 for the Ergodic model.

4.2. Test

The test is an important step because it can involve the choice of characteristics (primitives) or the choice of the training method. Once the training was done, we evaluated the valid system using 10,000 images from the MNIST test database (see Table 2). The characteristics extracted from this database are passed to the two models: ergodic HMMs and left-right HMMs. The size of the Codebook used is 232.

The results are given in the result and discussion part.

4.3. Recognition

The image to be recognized is first prepared by the preprocessing phase, then zoned into 09 zones to extract the characteristics to obtain a vector which will be used by the two types of HMM model to calculate the logarithm of likelihood of the figure to be recognized by report to all models, and we will assign this image to the class including the HMM that gives us the greatest probability of belonging. The schematic of the proposed system recognition system is shown in figure 10 below:

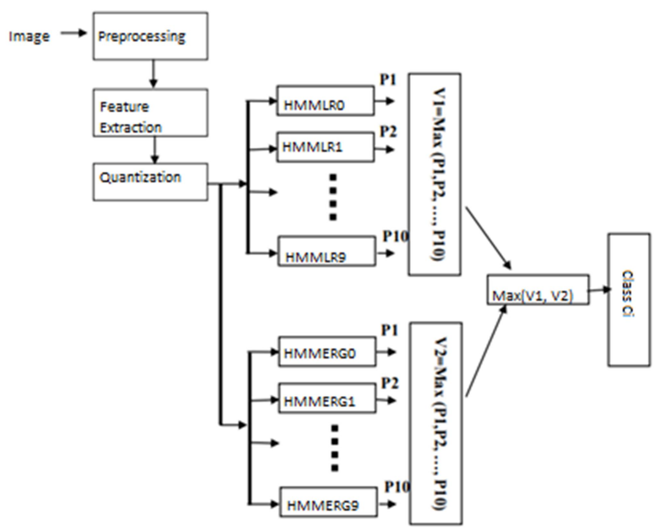

Fig. 10 The diagram of proposed recognition system

5. Results and discussions

To validate the proposed approach, we carried out experiments on two databases of MNIST handwritten digits, one relating to the training of 20,000 images in size for training and model validation, the other for testing size 10,000 images to evaluate the proposed system.

Here is the definition of a recognition rate: $Recognition\ Rate = \frac{Number\ of\ Recognized\ Images}{Total\ Number\ of\ Images} \times 100$

5.1. The results of the experiments carried out on the database of MNIST validation

The models were trained on 20,000 images from the MNIST database. Table 3 below indicates the results obtained after a training process carried out on the database of validation using the two models:

Table 3 The Max recognition rate reached by type of HMM and by the codebook

Codebook size	Recognition rate Left-Right HMM	Recognition rate Ergodic HMM
128	77.85	67.61
190	83.86	72.06
232	87.03	74.58

It can be seen from Table 3 that the size of the Codebook influences the recognition result. For a Codebook of size 232, the value of the recognition rate increases from 77.85% to 87.45% for the Left-Right HMM, respectively from 67.61% to 74.48% for the Ergodic HMM.

Table 4 below shows the results obtained after several operations (35 operations) carried out with different numbers of states:

Table 4 Results obtained with different status numbers (the size of the Codebook = 232)

State	Recognition Rate Left-Right HMM	Recognition Rate Ergodic HMM	State	Recognition Rate Left-Right HMM	Recognition Rate Ergodic HMM
4	67.85	62.21	22	81.61	72.55
5	72.26	72.10	23	82.67	69.14
6	70.02	71.58	24	81.16	70.59
7	74.28	69.36	25	83.30	73.22
8	74.03	68.56	26	83.04	72.87
9	75.16	71.92	27	77.77	72.42
10	76.25	71.52	28	84.29	70.13
11	75.88	71.33	29	85.90	71.39
12	77.85	73.25	30	76.45	70.63
13	77.20	71.30	31	84.19	69.06
14	77.85	72.87	32	84.99	72.02
15	79.45	72.43	33	85.25	70.71
16	79.41	70.78	34	85.90	73.31
17	79.54	72.45	35	86.38	69.60
18	81.49	73.08	36	78.40	72.10
19	73.61	72.99	37	79.28	73.12
20	80.77	69.64	38	77.13	70.52
21	80.54	70.08			

Figure 11 below shows that the probabilities of the two models converge towards a maximum value after 35 operations:

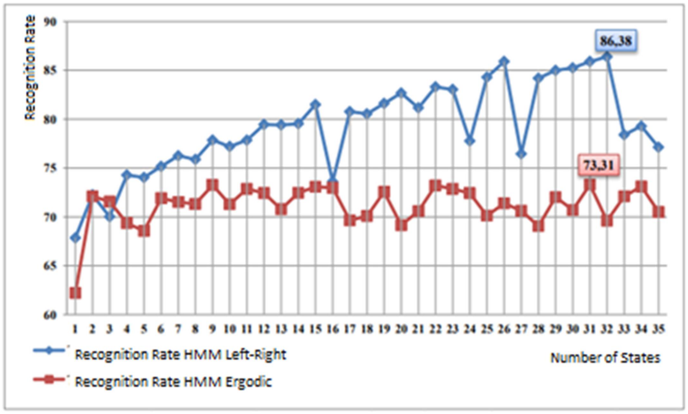

Fig. 11 comparative curve showing the relevance of the Left-Right HMM in the recognition of isolated digits compared to the Ergodic HMM

Thus, the number of 34 states seems to be best suited for the Ergodic model and 35 states for the Left-Right model. For each of the two models, we performed 13 iterations on the validation database with the two numbers of valid states (35 states for the Left-Right model and 34 states for the Ergodic model).

We notice that after 08 iterations on these two numbers of states, we obtained an improvement in the recognition rate of the two models. The results obtained are given in Table 5:

Table 5 The 13 iterations performed on the two Left-Right and Ergodic HMMs with a Codebook of size 232.

N° iteration	Recognition Rate Left-Right HMM (#states =35)	Recognition Rate Ergodic HMM (#states =34)
1	85.76	72.64
2	86.69	71.24
3	85.74	71.58
4	85.71	71.69
5	85.62	71.32
6	79.37	70.73
7	85.97	73.27
8	87.03	74.58
9	85.84	72.07
10	86.47	69.42
11	85.64	71.28
12	81.61	70.93
13	79.41	72.93

We achieve a recognition rate of:

- 87.03% for the first classifier (Left-Right HMM), the associated confusion matrix is given in Table 6 and Figure 12.
- 74.58% for the second classifier (Ergodic HMM), the confusion matrix is given in Table 7 and Figure 13.

Table 6 Confusion matrix corresponding to the first classifier (Left-Right based on the validation of MNIST of the model). The columns correspond to the recognized values and the rows to the true values (Number of states: 35).

Classes	0	1	2	3	4	5	6	7	8	9	Recognition Rate
0	1913	3	9	2	6	8	25	2	24	7	95.70
1	0	2044	34	8	14	15	15	43	6	0	93.80
2	14	14	1727	49	7	31	39	53	68	13	85.71
3	1	8	45	1685	10	97	20	35	97	52	82.20
4	9	20	17	18	1699	5	54	38	18	55	87.89
5	4	9	18	42	5	1652	25	11	28	23	90.92
6	24	27	53	19	21	49	1591	61	58	40	81.88
7	11	23	63	27	33	5	59	1881	12	26	87.90
8	6	11	27	67	33	45	34	23	1663	82	83.53
9	11	8	28	24	63	55	67	54	62	1561	80.76
Global Recognition Rate 87,03%											

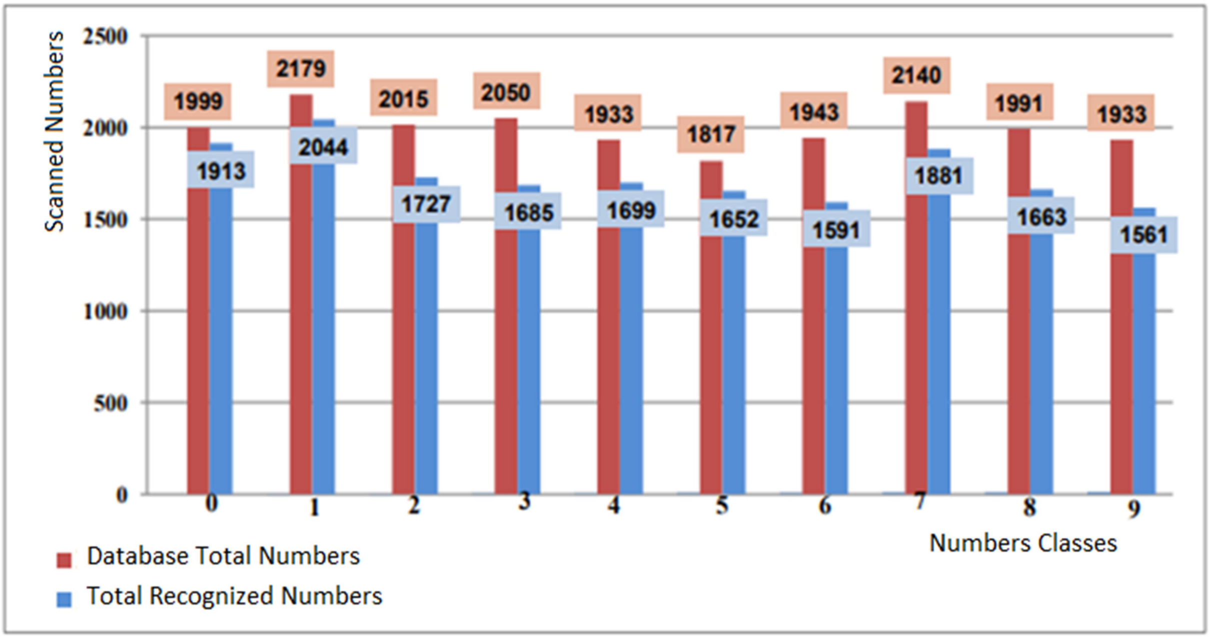

Figure 12 Histogram illustrating the confusion matrix of the first classifier (Left-Right model) based on validation with 35 states

Table 7 Confusion matrix corresponding to the second classifier (ergodic) based on MNIST validation of the model. The columns correspond to recognized values and the rows to true values (Number of states:34).

Classes	0	1	2	3	4	5	6	7	8	9	Recognition rate
0	1872	8	6	4	5	12	61	0	26	5	93.65
1	1	1969	38	11	29	57	22	43	6	3	90.36
2	50	36	1484	60	18	92	50	106	98	21	73.65
3	10	14	59	1160	17	480	23	48	189	50	56.59
4	11	45	23	22	1493	51	139	37	64	48	77.24
5	11	27	26	46	9	1582	24	12	28	52	87.07
6	64	54	57	19	49	141	1374	46	95	44	70.72
7	26	181	103	60	69	63	92	1441	58	47	67.34
8	25	29	52	77	33	151	42	24	1436	122	72.12
9	37	21	32	25	105	285	85	55	183	1105	57.17
Global Recognition Rate 74,59%											

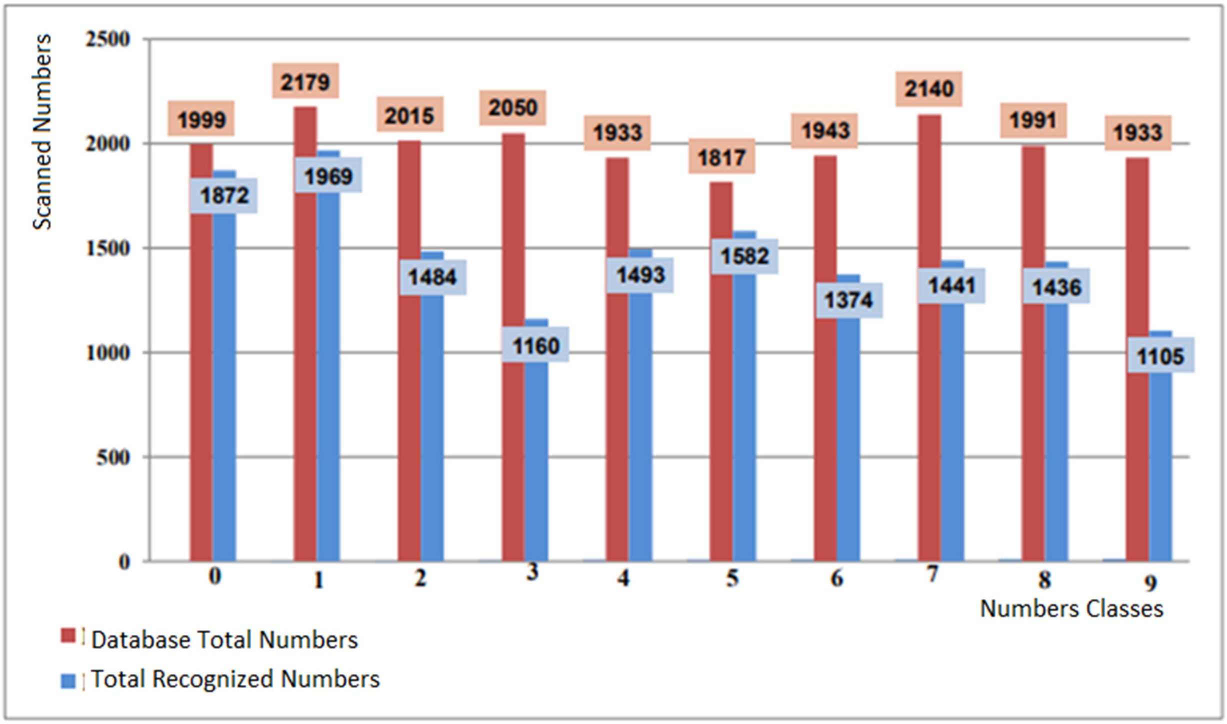

Figure 13 Histogram illustrating the confusion matrix of the second classifier (HMM_Ergodic) on the database of validation with 34 states

5.2. The results of the experiments carried out on the database of MNIST Test

The parameters valid for each model in the previous experiments were evaluated by tests carried out on 10,000 images of the MNIST database in five groups (See Table 8).

Table 8 MNIST test database recognition rate (by type of HMM)

HMM Topology	Codebook Size	State	Number of images (MNIST Test Base)				
			100	200	1000	5000	10000
Left-Right	232	35	74.00	67.00	63.60	62.96	64.47
Ergodic		34	77.00	72.00	67.80	67.54	67.89

The recognition results of the two models are summarized in Table 9 below:

Table 9 Recognition rate of the MNIST test database by group and by type of HMM

Classes	Number of images									
	100		200		1000		5000		10000	
	HMM L-R	HMM ERG	HMM L-R	HMM ERG	HMM L-R	HMM ERG	HMM L-R	HMM ERG	HMM L-R	HMM ERG
0	90.00	90.00	85.00	95.00	83.00	92.00	84.40	90.60	84.39	89.59
1	90.00	90.00	75.00	80.00	71.00	81.00	70.60	81.20	77.09	86.96
2	40.00	70.00	45.00	65.00	51.00	64.00	55.20	58.40	55.04	59.11
3	70.00	60.00	65.00	50.00	63.00	61.00	63.80	61.20	63.96	56.73
4	100.00	90.00	80.00	75.00	67.00	71.00	66.80	74.60	63.54	71.18
5	80.00	80.00	70.00	90.00	58.00	78.00	54.80	75.80	61.32	79.48
6	80.00	90.00	70.00	85.00	59.00	51.00	49.20	49.00	53.03	54.49
7	70.00	80.00	65.00	65.00	61.00	61.00	66.00	63.80	70.14	64.88
8	70.00	90.00	55.00	70.00	55.00	66.00	54.00	62.00	53.70	61.81
9	50.00	30.00	60.00	45.00	68.00	53.00	64.80	58.80	62.44	54.71
Global Rate	74.00	77.00	67.00	72.00	63.60	67.80	62.96	67.54	64.47	67.89

These results indicate:

- The recognition rate of digits: 0, 1, 2, 5, 6, 7.8 is better using the Ergodic HMM;
- The recognition rate of digits 3 and 9 is better using the HMM Left-Right;
- The recognition rate of the number 4 remains variable for the two models.

The experimental results obtained have shown that some errors of the first classifier can be avoided with the second classifier and vice versa. The reason why we propose the simultaneous use of the two models at the same time in order to increase the recognition rate by exploiting the performance of each of the two classifiers. Table 10 shows that the combined approach significantly improved the recognition rate.

Table 10 Comparative recognition rates of the Left-Right, Ergodic model and of the combined approach based on the MNIST test

HMM Topology	Codebook Size	State	Number of images (MNIST Test Base)				
			100	200	1000	5000	10000
Left-Right		35	74.00	67.00	63.60	62.96	64.47
Ergodic	232	34	77.00	72.00	67.80	67.54	67.89
Combined		35	81.00	75.50	70.30	68.64	69.92

The recognition rates by group of images obtained after the application of the combined approach are presented in Table 11 below:

Table 11 Recognition rate of the combined approach obtained on the database of MNIST test

Classes	Number of Images				
	100	200	1000	5000	10000
0	90.00	95.00	92.00	90.60	89.59
1	90.00	80.00	81.00	81.20	86.96
2	70.00	65.00	64.00	58.40	59.11
3	70.00	65.00	63.00	63.80	63.96
4	100.00	80.00	71.00	74.60	71.18
5	80.00	90.00	78.00	75.80	79.48
6	90.00	85.00	59.00	49.20	54.49
7	80.00	65.00	61.00	66.00	70.14
8	90.00	70.00	66.00	62.00	61.81
9	50.00	60.00	68.00	64.80	62.44
Global Rate	81.00	75.50	70.30	68.64	69.92

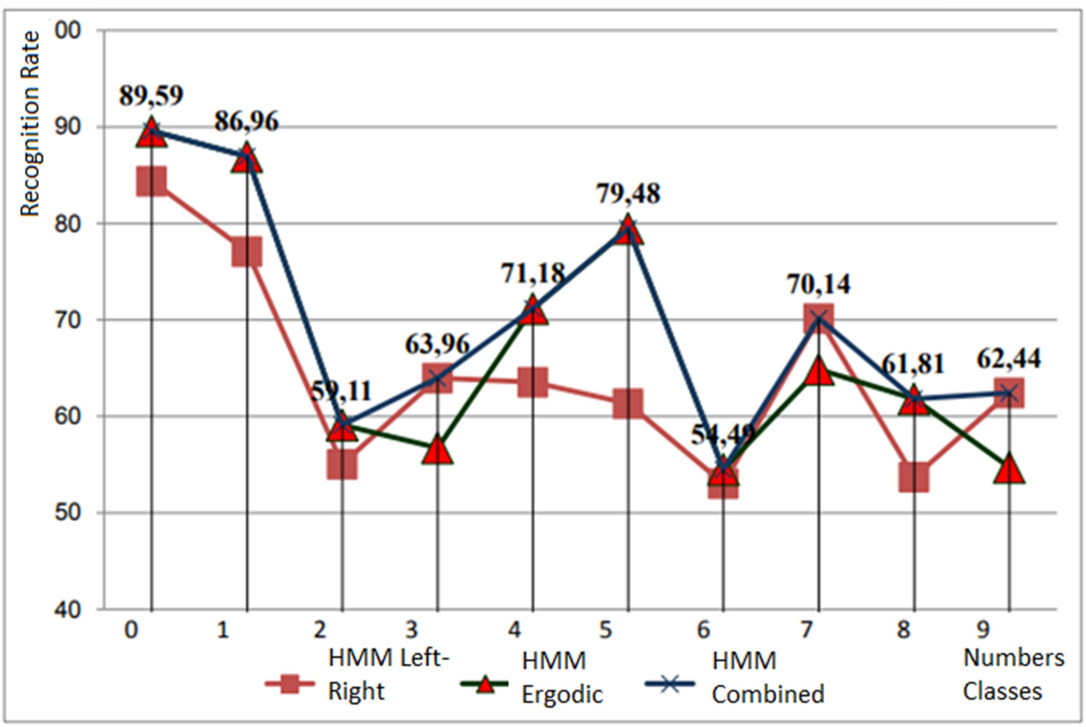

Figure 14 Histogram illustrating the confusion matrix of the combined approach (for the 10,000 images of the test base)

6. Conclusion and Future Work

The main objective of this study is to apply this Markov hidden model to recognition through the implementation of an automatic recognition system for handwritten digits. We first proposed to use the zoning method for feature extraction. Secondly, develop two discrete Markovian classifiers: the first of the Left-right type and the other of the Ergodic type.

The two models were trained on a database of the MNIST database of size 40,000 images in order to achieve a good training. The two classifiers implemented were tested on a large corpus of handwritten digits from the same database (MNIST) of around 10,000 images in order to test the real reliability of our system. We then proposed a combination of these two classifiers. This combination improved the recognition rates from 67.89% to 69.92% and reduced the error rate from 32.11% to 30.08%. The results obtained indicate that the simultaneous use of the two classifiers at the same time can make it possible to increase the recognition rate, which has not yet reached the best performances in the state of the art. However there are still tasks to be completed such as:

- Proposal to develop new methods for the extraction of characteristics.
- Adapt our system for other applications, such as character or word recognition.
- Combine several classifiers (more than two).

A major perspective is to cooperate the hidden Markov models with other classifiers such as neural networks or SVMs, which also remain a field open to experimentation.

References

[1] Cheriet, Mohamed, Nawwaf Kharma, Cheng-Lin Liu, and Ching Suen. Character recognition systems: a guide for students and practitioners. John Wiley & Sons, 2007.

[2] Schalkoff, Robert J. "Pattern recognition." Wiley Encyclopedia of Computer Science and Engineering (2007). https://doi.org/10.1002/9780470050118.ecse302

[3] Oliveira, Luiz S., Robert Sabourin, Flávio Bortolozzi, and Ching Y. Suen. "Automatic recognition of handwritten numerical strings: A recognition and verification strategy." IEEE Transactions on Pattern Analysis and Machine Intelligence 24, no. 11 (2002): 1438-1454. https://doi.org/10.1109/tpami.2002.1046154

[4] Hastie, Trevor, and Patrice Simard. "Metrics and models for handwritten character recognition." In Conference on Statistical Science Honouring the Bicentennial of Stefano Franscini's Birth, pp. 203-219. Birkhäuser, Basel, 1997. https://doi.org/10.1214/ss/1028905973

[5] LeCun, Yann, L. D. Jackel, Leon Bottou, A. Brunot, Corinna Cortes, J. Denker, Harris Drucker et al. "Comparison of learning algorithms for handwritten digit recognition." In International conference on artificial neural networks, vol. 60, pp. 53-60. 1995.

[6] Hu, Jianming, and Hong Yan. "Structural primitive extraction and coding for handwritten numeral recognition." Pattern Recognition 31, no. 5 (1998): 493-509. https://doi.org/10.1016/s0031-3203(97)00095-2

[7] Nishida, Hirobumi. "Curve description based on directional features and quasi-convexity/concavity." Pattern Recognition 28, no. 7 (1995): 1045-1051. https://doi.org/10.1016/0031-3203(94)00177-n

[8] Zou, Ju Jia, and Hong Yan. "Extracting strokes from static line images based on selective searching." Pattern Recognition 32, no. 6 (1999): 935-946. https://doi.org/10.1016/s0031-3203(98)00129-0

[9] Chong, Chee-Way, Paramesran Raveendran, and Ramakrishnan Mukundan. "Translation invariants of Zernike moments." Pattern recognition 36, no. 8 (2003): 1765-1773. https://doi.org/10.1016/s0031-3203(02)00353-9

[10] Hwang, Sun-Kyoo, and Whoi-Yul Kim. "A novel approach to the fast computation of Zernike moments." Pattern Recognition 39, no. 11 (2006): 2065-2076. https://doi.org/10.1016/j.patcog.2006.03.004

[11] Mahmoud, Sabri A. "Arabic character recognition using Fourier descriptors and character contour encoding." Pattern recognition 27, no. 6 (1994): 815-824.

[12] Kapogiannopoulos, George S., and Manos Papadakis. "Character recognition using a biorthogonal discrete wavelet transform." In Wavelet Applications in Signal and Image Processing IV, vol. 2825, pp. 384-393. International Society for Optics and Photonics, 1996.

[13] Gader, Paul, Brian Forester, Margaret Ganzberger, Andrew Gillies, Brian Mitchell, Michael Whalen, and Todd Yocum. "Recognition of handwritten digits using template and model matching." Pattern recognition 24, no. 5 (1991): 421-431. https://doi.org/10.1016/0031-3203(91)90055-a

[14] Jain, Anil K., and Douglas Zongker. "Representation and recognition of handwritten digits using deformable templates." IEEE Transactions on Pattern Analysis and Machine Intelligence 19, no. 12 (1997): 1386-1390. https://doi.org/10.1109/34.643899

[15] Lee, Dar-Shyang. "Handprinted digit recognition: A comparison of algorithms." In Proceedings of the Third International Workshop on Frontiers in Handwriting Recognition, pp. 153-164. 1993.

[16] Lee, Seong-Whan, Louisa Lam, and Chin Y. Suen. "Performance evaluation of skeletonization algorithms for document image processing." In Proc. 1 st Int. Conf. on Document Analysis and Recognition, vol. 1, pp. 260-271. 1991.

[17] Lee, Seong-Whan. "Off-line recognition of totally unconstrained handwritten numerals using multilayer cluster neural network." IEEE Transactions on Pattern Analysis and Machine Intelligence 18, no. 6 (1996): 648-652. https://doi.org/10.1109/34.506416

[18] Lee, Seong-Whan, and Sang-Yup Kim. "Integrated segmentation and recognition of handwritten numerals with cascade neural network." IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 29, no. 2 (1999): 285-290. https://doi.org/10.1109/5326.760572

[19] Ayat, Nedjem-Eddine, Mohamed Cheriet, and Ching Y. Suen. "Optimization of the SVM kernels using an empirical error minimization scheme." In International Workshop on Support Vector Machines, pp. 354-369. Springer, Berlin, Heidelberg, 2002. https://doi.org/10.1007/3-540-45665-1_28

[20] Byun, Hyeran, and Seong-Whan Lee. "Applications of support vector machines for pattern recognition: A survey." In International Workshop on Support Vector Machines, pp. 213-236. Springer, Berlin, Heidelberg, 2002. https://doi.org/10.1007/3-540-45665-1_17

[21] Micó, Luisa, and Jose Oncina. "Comparison of fast nearest neighbour classifiers for handwritten character recognition." Pattern Recognition Letters 19, no. 3-4 (1998): 351-356. https://doi.org/10.1016/s0167-8655(98)00007-5

[22] Cheung, Kwok-Wai, Dit-Yan Yeung, and Roland T. Chin. "A Bayesian framework for deformable pattern recognition with application to handwritten character recognition." IEEE Transactions on Pattern Analysis and Machine Intelligence 20, no. 12 (1998): 1382-1388. https://doi.org/10.1109/34.735813

[23] Cai, Jinhai, and Zhi-Qiang Liu. "Integration of structural and statistical information for unconstrained handwritten numeral recognition." IEEE Transactions on Pattern Analysis and Machine Intelligence 21, no. 3 (1999): 263-270. https://doi.org/10.1109/34.754622

[24] de S Britto, A., Robert Sabourin, Flavio Bortolozzi, and Ching Y. Suen. "A string length predictor to control the level building of HMMs for handwritten numeral recognition." In Object recognition supported by user interaction for service robots, vol. 4, pp. 31-34. IEEE, 2002. https://doi.org/10.1109/icpr.2002.1047393

[25] Procter, Steve, John Illingworth, and A. J. Elms. "The recognition of handwritten digit strings of unknown length using hidden Markov models." In Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No. 98EX170), vol. 2, pp. 1515-1517. IEEE, 1998. https://doi.org/10.1109/icpr.1998.711995

[26] Hanmandlu, Madasu, KR Murali Mohan, Sourav Chakraborty, Sumeer Goyal, and D. Roy Choudhury. "Unconstrained handwritten character recognition based on fuzzy logic." Pattern Recognition 36, no. 3 (2003): 603-623. https://doi.org/10.1016/s0031-3203(02)00069-9

[27] Shridhar, Malayappan, and A. Badreldin. "Recognition of isolated and simply connected handwritten numerals." Pattern Recognition 19, no. 1 (1986): 1-12. https://doi.org/10.1016/0031-3203(86)90025-7

[28] Hirano, Takashi, Yasuhiro Okada, and Fumio Yoda. "Structural character recognition using simulated annealing." In Proceedings of the fourth International conference on document Analysis and Recognition, vol. 2, pp. 507-510. IEEE, 1997. https://doi.org/10.1109/icdar.1997.620550

[29] Kang, Hee-Joong. "Combining multiple classifiers based on third-order dependency for handwritten numeral recognition." Pattern Recognition Letters 24, no. 16 (2003): 3027-3036. https://doi.org/10.1016/s0167-8655(03)00163-6

[30] Chmielnicki, Wiesław, and Katarzyna Stąpor. "Investigation of normalization techniques and their impact on a recognition rate in handwritten numeral recognition." Schedae Informaticae 19 (2010). https://doi.org/10.2478/v10149-011-0004-y

[31] Suen, Ching Y., Christine Nadal, Raymond Legault, Tuan A. Mai, and Louisa Lam. "Computer recognition of unconstrained handwritten numerals." Proceedings of the IEEE 80, no. 7 (1992): 1162-1180.

[32] Teow, Loo-Nin, and Kia-Fock Loe. "Robust vision-based features and classification schemes for off-line handwritten digit recognition." Pattern Recognition 35, no. 11 (2002): 2355-2364. https://doi.org/10.1016/s0031-3203(01)00228-x

[33] Belongie, Serge, Chad Carson, Hayit Greenspan, and Jitendra Malik. "Recognition of images in large databases using a learning framework." UC Berkeley CS Technical Report (1997): 97-939.

[34] Dong, Jian-xiong, Adam Krzyżak, and Ching Y. Suen. "An improved handwritten Chinese character recognition system using support vector machine." Pattern Recognition Letters 26, no. 12 (2005): 1849-1856. https://doi.org/10.1016/j.patrec.2005.03.006

[35] LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86, no. 11 (1998): 2278-2324. https://doi.org/10.1109/5.726791

[36] Blumenstein, Michael, Brijesh Verma, and Hasan Basli. "A novel feature extraction technique for the recognition of segmented handwritten characters." In Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings., pp. 137-141. IEEE, 2003. https://doi.org/10.1109/icdar.2003.1227647

[37] Baum, Leonard E. "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes." Inequalities 3, no. 1 (1972): 1-8.

[38] Baum, Leonard E., and Ted Petrie. "Statistical inference for probabilistic functions of finite state Markov chains." The annals of mathematical statistics 37, no. 6 (1966): 1554-1563. https://doi.org/10.1214/aoms/1177699147