Intelligent Automation & Soft Computing

ISSN: 1079-8587 (Print) 2326-005X (Online) Journal homepage: http://www.tandfonline.com/loi/tasj20

Comparing the machine ability to recognize hand-written Hindu and Arabic digits

Khalil El Hindi, Muna Khayyat & Areej Abu Kar

To cite this article: Khalil El Hindi, Muna Khayyat & Areej Abu Kar (2016): Comparing the machine ability to recognize hand-written Hindu and Arabic digits, Intelligent Automation & Soft Computing, DOI: 10.1080/10798587.2016.1210257

To link to this article: http://dx.doi.org/10.1080/10798587.2016.1210257

Published online: 26 Jul 2016.

INTELLIGENT AUTOMATION & SOFT COMPUTING, 2016
http://dx.doi.org/10.1080/10798587.2016.1210257

Comparing the machine ability to recognize hand-written Hindu and Arabic digits

Khalil El Hindi[a], Muna Khayyat[b] and Areej Abu Kar[c]

[a]Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia; [b]Computer Science and Software Engineering Department, Concordia University, Montreal, Canada; [c]King Abdullah II School for IT, University of Jordan, Amman, Jordan

ABSTRACT
The main aim of this work is to compare Hindu and Arabic digits with respect to a machine's ability to recognize them. This comparison is done on the raw representation (images) of the digits and on their features extracted using two feature selection methods. Three learning algorithms with different inductive biases were used in the comparison performed using the raw representation; two of them were also used to compare the digits using their extracted features. All classifiers gave better results for Hindu digits in both cases; when raw representation was used and when the selected features where used. The experiments also show that Hindu digits can be classified with better accuracy, higher confidence and using fewer features than Arabic digits. These results indicate that hand-written Hindu digits are actually easier to recognize than hand-written Arabic digits. The machine learning methods used in this work are instance based learning (the kNN algorithm), Naïve Bayesian and neural networks. The feature extraction methods we used were Fourier transformation and histograms.

KEYWORDS
Hand-written digit recognition; feature extraction; Instance-based learning; neural network; Bayesian classifiers

1. Introduction

Handwritten digit recognition is one of the most successful applications of automatic pattern recognition. However, most of the handwritten digit recognition applications were applied on Arabic digits, since it is the most popular numbering system in the world. Surprisingly, in the Arab world, Hindu digits are much more widely used in everyday life than Arabic digits, except in some North African Arab countries such as Morocco and Algeria, where only Arabic digits are used. However, there is a call among some Arab intellectuals to use Arabic digits in everyday life and abandon the Hindu representation, to be in line with the rest of the world. Furthermore, some claim that Arabic digits are easier to recognize than Hindu digits. The aim of this study is to compare Hindu and Arabic representations of digits in attempt to determine which representation is easier for a machine to recognize. Table 1 shows a sample of hand-written Arabic and their corresponding Hindu digits.

In recent years many attempts were made to recognize Hindu digits (Alfonse, Almorsy, & Barakat, 2010; Al-Omari & Al-Jarrah, 2004; Mahmoud & Awaida, 2009; Zaghloul, Bader, & Al, 2012). However, their aim was to achieve maximum classification accuracy for Hindu digits and as far as we know, no attempt has been made to compare the Hindu and Arabic representation of digits with respect to a machine ability to recognize them.

The aim of this study is to compare Hindu and Arabic representations of digits in an attempt to determine which representation is easier for a machine to recognize.

Since machine learning algorithms have different biases, it is vital to compare the two representations using different machine learning algorithms. The inductive bias of an algorithm is what makes it able to generalize and classify new unseen instances (Mitchell, 1997). Due to inductive biases, there is no machine learning algorithm that is superior to all other algorithms in all applications (Schaffer, 1994). That is why we chose to use three machine learning algorithms in our comparison. The three algorithms represent different approaches in machine learning. The algorithms are the Naïve Bayesian algorithm (Domgos & Pazzani, 1996; Duda & Hart, 1973), as a representative of the statistical approaches; the k Nearest Neighbor algorithm (Aha, Kibler, & Albert, 1991; Stanfill & Waltz, 1986) as a representative of Instance Based Learning (IBL); and the RPROP algorithm (Riedmiller & Braun, 1993) as a representative of Artificial Neural Network (ANN) training algorithms.

In the empirical study, two sets of experiments were performed. In the first, the raw images of the digits were used, while in the second sets extracted features from these images were used. Feature extraction is based on extracting information from the raw digit images that are relevant for classification. The aim is to minimize a within-class pattern variability while enhancing the between class pattern variability (Devijver & Kittler, 1982).

In this work, two feature selection approaches are used: Fourier transformation and histograms. These are general purpose feature extraction methods, that are supposed to perform equally well on Arabic and Hindu digits.

It is important to stress that this work does not aim at comparing the learning algorithms but rather the Arabic and Hindu digits. Therefore, it is sufficient for our purpose to compare the Hindu and Arabic digits using the same algorithms and same parameter values and not necessarily the best parameter values.

Section 2 reviews the machine learning algorithms used in this work. Section 3 discusses the feature selection methods. Section 4 describes the experiments performed and presents the results. Section 5 is the conclusion.

CONTACT Khalil El Hindi khindi@ksu.edu.sa

Table 1. A sample of handwritten Arabic and Hindu digits.

Arabic	0	1	2	3	4	5	6	7	8	9
Hindu	٠	١	٢	٣	٤	٥	٦	٧	٨	٩

2. The machine learning algorithms used

This section describes the three learning algorithms used in this work, namely; Naïve Bayesian, Instance-Based Learning, and the RPROP algorithm.

2.1. Naïve Bayesian classifier

The Naïve Bayesian approach to learning (Domgos & Pazzani, 1996; Duda & Hart, 1973) uses the Bayesian rule for conditional probabilities. To classify a new instance, the method calculates the probability of each class value, given the values of the instance's attributes. The class with the maximum probability is then taken as the predicted class of the new instance. The training set is used to estimate all needed probabilities.

Given the vector of attribute values $< a_1,a_2,\ldots,a_n >$, the probability for a class value, C, is computed as:

$$P(C|a_1,a_2,\ldots,a_n) = \frac{P(a_1,a_2,\ldots,a_n|C)P(C)}{P(a_1,a_2,\ldots,a_n)} \tag{1}$$

where: $P(C)$ is the probability of class C; $P(a_1,a_2,\ldots,a_n)$ is the probability that the attributes $1,2,\ldots,n$ will take the values $a_1,a_2,\ldots,a_n$, respectively; $P(a_1,a_2,\ldots,a_n|C)$ is the probability that the attributes $1,2,\ldots,n$ will take the values $a_1,a_2,\ldots,a_n$ given that the instance is of class C.

Since, given a certain instance, the probability $P(a_1,a_2,\ldots,a_n)$ is the same, therefore Formula 1 can be simplified as follows:

$$P(C|a_1,a_2,\ldots,a_n) = P(a_1,a_2,\ldots,a_n|C)P(C) \tag{2}$$

The approach got its name because it naively assumes that attribute values are conditionally independent given the class value. Therefore, it assumes that:

$$P(a_1,a_2,\ldots,a_n|C) = \prod_i P(a_i|C) \tag{3}$$

2.2. The k Nearest Neighbor algorithm

Instance-Based learners (IBL) (Aha et al., 1991; Stanfill & Waltz, 1986), such as the k Nearest Neighbor (kNN) algorithm, are machine-learning methods that retain training examples in an instance memory. To classify a new (previously unseen) instance, the method retrieves the k most similar instance (its k nearest neighbors). It uses the most frequent class of the retrieved instances as the predicted class for the new instance.

To measure the similarity between instances, a similarity (distance) function is used (Stanfill & Waltz, 1986; Wilson & Martinez, 1997; El Hindi, 2013). The similarity function used in this work is a variant of the HVDM (Wilson & Martinez, 1997). It is defined as:

$$HVDM(X,Y) = \sqrt{\sum_{a=1}^{m} d_a^2(x_a,y_a)} \tag{4}$$

where X and Y are two instances, m is the number of attributes, x_a and y_a are the values of attribute a in instances X and Y, respectively. The distance function, d_a, between two values depends on the type of the attribute (symbolic or numeric):

$$d_a(x,y) = \begin{cases} vdm_a(x,y), & \text{if } a \text{ is symbolic} \\ \frac{|x-y|}{a_{\max}-a_{\min}} & \text{if } a \text{ is numeric} \end{cases} \tag{5}$$

where *vdm* is the value distance metric (Stanfill & Waltz, 1986), defined as follows:

$$vdm_a(x,y) = \sum_{c=1}^{C}\left(\frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}}\right)^2 \tag{6}$$

where: a_{max} and a_{min} are the maximum and minimum values of attribute a; Na,x is the number of instances in the training set that have the value x for attribute a; Na,x,c is the number of instances in the training set of class C that have value x for attribute a; C is the number of classes in the problem domain.

2.3. The RPROP algorithm

The RPROP (Resilient PROPagation) algorithm (Riedmiller & Braun, 1993) differs from the backpropagation algorithm (Rumelhart & McClelland, 1986), in that it only uses the sign of the partial derivative of the weight to decide on the way it updates weights. The magnitude of weight update depends on the sign of consecutive derivatives. If the derivative does not change sign, the value of weight's update is increased. If it changes sign that means it has probably jumped over a local minima, and therefore, the value of the weight update is decreased.

We chose to use the RPROP algorithm in this work not only because it is faster than Backpropagation but also because it is less sensitive to the choice of parameters than the Backpropagation algorithm (Riedmiller & Braun, 1993). This makes the results we obtain in comparing Hindu and Arabic digits more reliable and less sensitive to the chosen parameter values.

3. Feature selection methods

In this section, we give a brief description of the two feature selection methods we used in this work, namely, Fourier transformation and histograms.

3.1. Fourier transformation

The Discrete Fourier Transform (DFT) (Kammler, 2000) is used in this work for extracting features from handwritten digits. DFT makes it possible to remove redundant information. It is based on the observation that most of the energy that lies in the low frequency component may be redundant due to the correlation between pixels (Moharir, 1992). Fourier Transformation is used in the literature as a feature selection method for character recognition (O'Hair, 1990; Shartle Gary, 1993).

In this work a, a shift functionality of a 2-dimensional FFT to grayscale-handwritten digits is employed.

3.2. Histograms

Horizontal and vertical histograms were found useful in optical character recognition (Kavallieratou, Sgarbas, Fakotakis, & Kokkinakis, 2003). They show the distribution of data values by binning elements into a number of equally spaced containers.

In this work, three histograms were used, namely; horizontal, vertical, and distance histograms.

Consider that the value of the element in the mth row and nth column of a digit matrix, given by the function

$$f(m, n) = a_{mn} \tag{7}$$

where a_{mn} takes a binary value (i. e., 0 for white pixels and 1 for black pixels). The horizontal histogram Hh of the digit matrix is the sum of black pixels in each row,

$$H_h(m) = \sum f(m, n) \tag{8}$$

Similarly, the vertical histogram Hv of the digit matrix is the sum of black pixels in each column

$$H_v(n) = \sum f(m, n) \tag{9}$$

Distance histogram is used to bin the means of the vertical and horizontal black pixels on the image to establish a matrix of distances.

4. Experiments and results

A group of 400 volunteers took part in building two datasets. Each volunteer were asked to write the digits (0–9) in both representations (Hindu and Arabic), ten times. The volunteers were selected from different ages, and educational levels. As a result, 8,000 written digits were gathered to generate two datasets, Hindu and Arabic, each of 4,000 written digits. In addition, different pens with different thicknesses were used giving more variability to the data. These written digits were scanned as 60 by 60 grayscale .jpg images.

Ten-fold cross validation was used in all experiments. In each experiment, 90% of each data-set was used in training the algorithm (as a training data) and the remaining 10% was used as a test set. This was repeated 10 times for each algorithm, replacing each time the 10% test set with another 10% of the training set.

4.1. Results using raw images

In the following subsections, we discuss the results we obtained using each algorithm when trained on the raw images of digits.

4.1.1. The results of the naïve Bayesian

The obtained average classification accuracy of the Naïve Bayesian algorithm on the Arabic and Hindu digits is 60.93%, 83.75%, respectively. This is a considerable difference that merits further analysis. Figure 1 shows the classification accuracy for each Hindu and Arabic digit. It can be noticed that the Hindu digit, with the poorest classification accuracy, is digit 9 with a classification accuracy of 72.5% while its corresponding Arabic digit has even a lower classification accuracy of 70.25%. However, the Arabic digit with the poorest classification accuracy is digit 5, with 38.25% average classification accuracy, while its Hindu counterpart has a much higher classification accuracy of 84%.

Table 2 is the confusion matrix, which shows for each Arabic digit, the number of times the algorithm mistakenly classified it as another digit. The matrix may help give an idea about how similar the Naive Bayesian algorithm finds the Arabic digits to each other. Table 3 is the corresponding confusion matrix for Hindu digits.

Table 2 shows that Arabic digit 3 was mistakenly classified as 7 sixty-two times, and digit 5 was mistakenly classified as 6 fifty-eight times, and digit 2 was mistakenly classified as 1 fifty-seven times. Table 3 shows that the most confusing pair of Hindu digits is digits 3 and 7; digit 3 was misclassified as 7 thirty-one times and digit 7 was misclassified as 3 fifty-nine times. It is obvious from the tables that the Naïve Bayesian algorithm

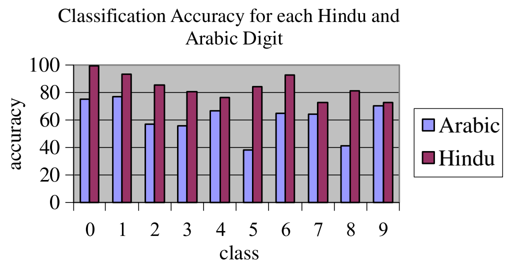

Figure 1. Classification accuracy for each Arabic and Hindu digit obtained using the Naïve Bayesian algorithm.

Table 2. The confusion matrix of Arabic digits using Naïve Bayesian algorithm.

		Misclassified as									
		0	1	2	3	4	5	6	7	8	9
Input digit	0		35	3	0	0	11	16	7	4	27
	1	1		37	8	3	5	25	10	3	3
	2	27	57		13	6	5	11	41	9	6
	3	16	39	27		0	5	2	62	14	12
	4	22	34	5	3		8	7	5	16	34
	5	47	28	8	37	5		58	30	11	23
	6	39	51	14	2	0	4		17	14	0
	7	20	17	29	43	3	3	7		6	15
	8	34	45	34	27	2	11	26	28		28
	9	20	27	9	8	12	4	3	17	19	

Table 3. The confusion matrix of Hindu digits using Naïve Bayesian algorithm.

		Misclassified as									
		0	1	2	3	4	5	6	7	8	9
Input digit	0		0	0	0	0	0	1	0	2	0
	1	3		1	4	12	0	0	7	1	0
	2	4	1		8	7	17	1	3	7	9
	3	0	0	24		5	4	0	31	13	2
	4	2	23	31	3		2	0	16	2	16
	5	3	0	13	1	4		1	12	23	7
	6	2	0	3	0	0	2		2	14	7
	7	4	0	0	59	2	13	19		8	3
	8	3	1	1	0	13	38	0	11		9
	9	2	0	2	1	26	30	3	3	43	

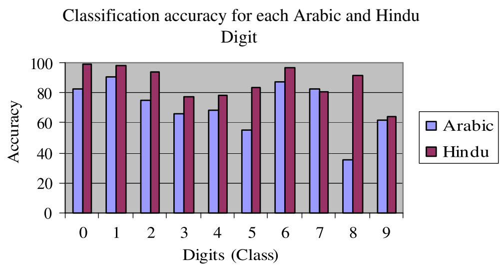

Figure 2. Classification accuracy for each Arabic and Hindu digit using Nearest Neighbor algorithm.

finds more confusing pairs of Arabic digits than Hindu digits. In other words, Hindu digits are easier to distinguish between.

We used another criterion to compare the level of difficulty of the two digit representation methods. It is the level of confidence that the algorithm has in its classification results. We used the difference between the probability of the predicted digit (the digit with the highest probability) and the probability of the digit with the second highest probability, as an indicator to the level of confidence in the classification result. The larger this difference is, the higher the confidence in the produced result is. The average differences in the Hindu and Arabic cases are 6.66E-72 and 3.70E-73, respectively. This shows that the Naïve Bayesian algorithm is more confident in its classification result in the Hindu case, since the average confidence is much higher. This, also, indicates that the algorithm finds Hindu digits easier to classify.

4.1.2. The results of the kNN algorithm

The average classification accuracies obtained using the kNN algorithm on Arabic and Hindu digits were 70.65% and 86.53%, respectively. Again a considerable difference in favor of the Hindu digits. Figure 2 shows that all Hindu digits have higher classification accuracies than their Arabic counterparts, except for Hindu digit 7, which has a slightly smaller classification accuracy than its Arabic counterpart. The accuracy of Hindu digit, 7, is 81%, while the accuracy of its Arabic counterpart is 82.25%. The Hindu digit with the poorest classification accuracy is digit 9 with 64% accuracy. However, it's Arabic counter- part has even smaller accuracy of 62%. On the other hand, the Arabic digit with the poorest classification accuracy is digit 8 with classification accuracy of 35.5%, while, its Hindu counterpart has a much higher accuracy of 92%. Table 4 and Table 5 show the confusion matrices for the Arabic and Hindu digits, respectively. A quick look at these tables show that the

Table 4. The confusion matrix of Arabic digits using Nearest Neighbor algorithm.

		Misclassified as									
		0	1	2	3	4	5	6	7	8	9
Input digit	0		10	2	1	1	1	46	0	0	8
	1	1		7	0	10	2	4	10	0	2
	2	8	22		14	1	5	1	16	11	21
	3	3	17	22		0	28	4	50	4	7
	4	10	69	11	0		0	25	3	0	8
	5	13	75	9	23	10		26	11	4	7
	6	14	25	4	0	0	1		5	0	0
	7	4	20	11	12	9	8	2		1	4
	8	36	22	50	29	5	41	22	13		40
	9	94	4	6	7	24	7	4	5	2	

Table 5. The confusion matrix of Hindu digits using Nearest Neighbor algorithm.

		Misclassified as									
		0	1	2	3	4	5	6	7	8	9
Input digit	0		1	0	0	0	1	1	0	0	0
	1	4		0	0	0	0	0	0	0	1
	2	3	11		6	0	1	1	0	0	0
	3	3	58	3		0	1	0	24	0	1
	4	16	14	40	7		3	1	3	0	1
	5	4	5	23	0	4		4	1	24	2
	6	10	0	3	0	0	0		1	0	0
	7	27	37	4	0	1	0	3		2	2
	8	7	10	0	0	0	1	10	4		0
	9	5	0	6	1	1	70	37	8	17	

algorithm finds many pairs of Arabic digits difficult to distinguish between compared to the Hindu digits.

We used the distance to the nearest enemy, as a measure of the level of confidence in the obtained classifications. The nearest enemy of an instance is the nearest instance that belongs to a different class. We computed the average distance to the nearest enemy for Hindu and Arabic digits and found that it is 6.397 for the Hindu and 4.385 for Arabic digits. This result shows that the nearest enemy was on average farther away for Hindu digits than it is for Arabic digits. This means that one can be more confident in the classifications results in the Hindu case.

4.1.3. The results obtained using neural networks

Two feed forward neural networks each with a hidden layer of 50 neurons were trained using the RPROP algorithm, one on the Hindu digits and one on the Arabic digits. They were trained for the same number of epochs. We did not try different network structures and parameter values, because our objective was not to compare the classification accuracy of the learning algorithms but rather to compare the Hindu and Arabic representations of digits. Therefore, the important thing for our purpose is to use the same network structure and parameter values. The classification accuracy we obtained for Arabic and Hindu digits are 63.2% and 81.4%, respectively. Once again, the classification accuracy of Hindu digit is far better than the classification accuracy of Arabic digits. Figure 3 shows the classification accuracy for each Hindu and Arabic digit. ables 6 and 7 are the confusion matrices for Arabic and Hindu digits, respectively.

As a measure of confidence in the classification result of the neural networks, we used the difference between the activation values of the two output neurons with the highest values. The higher the difference is the more confident one can be in the classification result. We found that the average difference of all Arabic and Hindu digits, on the test data, to be 0.446 and 0.689, respectively. This indicates that one can be more confident in classification accuracy of the neural network in the Hindu case than in the Arabic case.

4.2. The results obtained using features selection methods

This section discusses the classification results of Hindu and Arabic digits, obtained using two feature selection methods; Fourier Transformation, and Histograms. The input to the machine learning algorithms here is not the raw digit images, but rather the output of the feature selection algorithm.

4.2.1. Fourier transformation

A neural network of one hidden layer of thirty neurons was trained to classify Arabic Digits and a similar one was trained to classify Hindu Digits. The classification accuracy obtained

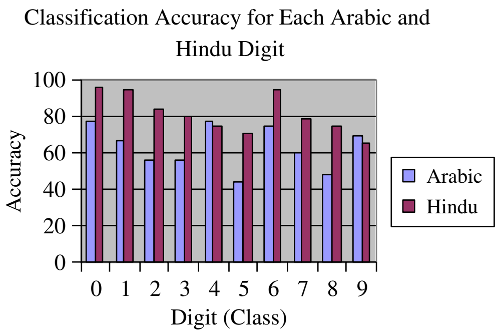

Figure 3. Classification accuracy for each Arabic and Hindu digit using RPROP.

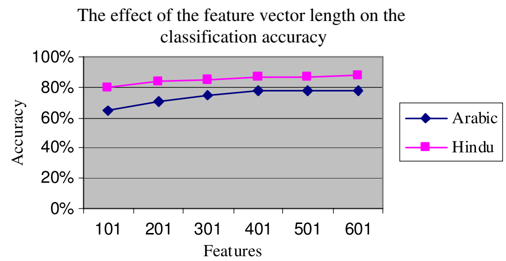

Figure 4. The effect of the feature vector length on the classification accuracy using Fourier transformation with the Back Propagation algorithm.

Table 6. The confusion matrix of Arabic digits using the RPROP algorithm.

		Misclassified as									
		0	1	2	3	4	5	6	7	8	9
Input digit	0		10	4	0	9	13	15	4	7	29
	1	8		37	17	9	18	20	5	19	1
	2	13	11		35	6	35	7	33	18	15
	3	3	10	42		7	20	1	52	31	7
	4	3	4	9	7		14	24	2	14	12
	5	16	11	15	41	16		26	39	47	13
	6	9	20	3	6	17	23		2	20	2
	7	4	3	42	49	9	22	7		18	5
	8	10	7	33	38	8	51	22	21		15
	9	29	7	8	17	11	15	5	5	25	

Table 7. The confusion matrix of Hindu digits using the RPROP algorithm.

		Misclassified as									
		0	1	2	3	4	5	6	7	8	9
Input digit	0		3	3	0	1	3	3	1	0	0
	1	0		1	1	9	2	1	6	4	0
	2	5	0		3	13	27	3	6	1	4
	3	0	7	15		12	9	2	24	5	5
	4	6	14	13	12		7	2	26	8	12
	5	0	1	20	7	8		3	7	41	31
	6	7	0	2	1	1	1		6	5	0
	7	2	17	0	31	6	5	5		14	5
	8	1	8	0	14	11	26	4	17		20
	9	0	0	3	10	6	72	5	13	30	

Table 8. The classification accuracy of the Arabic and Hindu digits using feature vectors of different lengths and distributions.

h_x	h_y	h_d	Length	Arabic%	Hindu%
25	25	101	151	75.4	84.9
50	50	51	151	81	85.9
50	50	21	121	81.4	86.7
25	25	51	101	78.2	86.2
30	30	41	101	82.4	86.9

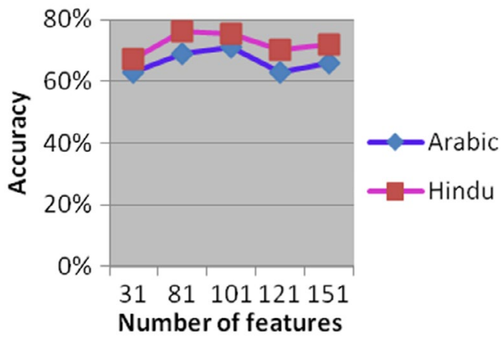

Figure 5. The classification accuracy of the kNN algorithm on Hindu and Arabic digits using different lengths for the feature vector.

using a feature vector of 601 features, is 77.8% for Arabic digits and 87.6% for Hindu digits, once again a considerable difference in favor of Hindu digits. Since the feature vector length is tunable, it is important to investigate the effect of altering the vector length on the classification accuracy. Figure 4 shows the classification accuracies obtained using different lengths of the feature vector. Note that, as the length of the feature vector decreases; the classification accuracy slowly drops down for Arabic and Hindu digits alike. However, the classification accuracy of the Hindu digits remains consistently higher than the classification accuracy of Arabic digits.

The classification accuracy obtained using the kNN algorithm, with k=3, and a feature vector of 601 features, is 84.8% for Arabic digits and 87.9% for Hindu digits. Another experiment was performed with a vector 501 features. The classification accuracies obtained were 77.8% and 87.5% for Arabic and Hindu digits, respectively, a drop of 7 points in the Arabic case and of only 0.4% in the Hindu case. This indicates that the Hindu digits can be classified reasonably well using fewer features than the Arabic digits.

4.2.2. Histograms

Another set of experiments were performed on the features extracted using the Histogram method described earlier.

Two neural networks, with one hidden layer of 70 neurons each, were trained, one on Arabic digits and one on Hindu digits. The experiment was repeated several times using different number of features. The classification accuracy results are presented in Table 8, where h_x and h_y are the horizontal and vertical histogram and h_d is the distance histogram. Table 8 presents the classification accuracy for Arabic and Hindu digits using different feature-vector lengths and different distributions of features. For example, using a feature vector of 151 features, with 25 features from the horizontal and the vertical histograms and 101 features from the distance histogram, gives a classification accuracy of 75.4% for Arabic digits and 84.9% for Hindu digits. While, a feature vector of the same length, but with 50 features from each of the horizontal and vertical histograms and 51 features from the distance histogram, gave an accuracy of 81% on Arabic digits and 85.9% on Hindu digits. However, the classification accuracy of the Hindu digits was always better than the classification accuracy of the Arabic digits in all experiments that we performed using different vector lengths and distributions of features, as shown in Table 8.

The kNN algorithm, with k=3, was also used on features extracted using the histogram method. Again, different feature vector lengths were used. Figure 5 shows the obtained classification accuracies for different vector lengths: 31, 81, 101, 121, and 151. For example, using a feature vector of 101 features, the algorithm gave classification accuracies of 75.5% and 71% on Hindu and Arabic digits, respectively. While, using a feature vector of 81 features, the algorithm gave classification accuracies of 75% on Hindu digits and 69% on Arabic digits. As can be easily seen in Figure 5, the algorithm gave better results on Hindu digits regardless of the length of the feature vector used.

5. Conclusion

This work compared the Arabic and Hindu representations of digits, with respect to a machine's ability to recognize them. Three different algorithms were used to ensure different inductive biases: Naïve Bayesian, kNN, and the RPROP algorithm.

It turned out that Hindu digits are easier to recognize. The empirical results reported in this work support this conclusion in three ways. All used algorithms gave better classification accuracies for Hindu digits than Arabic digits. This was true in both cases, when the algorithms were trained on the raw digit images (with no feature selection) and when they were trained on the extracted features using two feature selection methods: Fourier Transformation and Histograms.

Furthermore, the algorithms were more confident in their classification results in the Hindu case than the Arabic case. The third evidence, that supports our conclusion, is the fact that we obtained reasonable classification accuracy for Hindu digits with small feature vectors. Reducing the size of the feature vector resulted in a large reduction in the classification accuracy for Arabic digits. This indicates that Hindu digits can be classified using fewer features than Arabic digits.

Therefore, perhaps the call of some Arab intellectuals to abandon Hindu digits and use the Arabic digits should be reversed as a message to the rest of the world to use Hindu digits instead of Arabic digits, especially for applications that may require off-line recognition of digits.

The focus of this paper was on off-line handwritten digit recognition. It may be interesting to make a similar comparison using on-line handwritten digits and using other classification algorithms.

Acknowledgment

This work was supported by the Research Center of College of Computer and Information Science, King Saud University. The authors are grateful for this support.

Disclosure statement

No potential conflict of interest was reported by the authors.

Notes on contributors

Khalil El Hindi is an associate professor in the Department of Computer Science, King Saud University. His research interests include machine leaning and data mining. He is interested in improving the classification accuracy of Bayesian classifiers and developing new similarity metrics. He received his BS.C. degree from Yarmouk University and his M.Sc. and Ph.D. degrees from the University of Exeter, UK.

Muna Khayyat is a researcher at CENPARMI (Centre of Pattern Recognition and Machine Learning) in addition to her work in industry. Her research interests are in handwriting recognition and word spotting. She received Bachelor's degree from Birzeit University, Master's degree from the University of Jordan. and Ph.D. student at the Computer Science and Software Engineering Department of Concordia University.

Areej Abu kar is a consultant at PwC. She received a Bachelor Degree and Master Degree in Computer Science from the University of Jordan. Besides working in the industry, she is interested in machine learning and artificial intelligence related topics and research.

References

Aha, D., Kibler, D., & Albert, M. (1991). Instance-based learning algorithms. *Machine Learning, 6*, 37–66.

Alfonse, M., Almorsy, M., & Barakat, M.S. (2010). Eastern Arabic handwritten numerals recognition. *Int. J. Comput. Elect. Eng., 2*, 1793–8163.

Al-Omari, F.A., & Al-Jarrah, O. (2004). Handwritten Indian numerals recognition system using probabilistic neural networks. *Advanced Engineering Informatics, 18*, 9–16. doi:10.1016/j.aei.2004.02.001.

Devijver, P.A., & Kittler, J.. (1982). *Pattern recognition: A statistical approach*. New Jersey: Prentice Hall International.

Domgos, P., Pazzani, M. (1996). Beyond independence: Conditions for the optimality of the simple bayesin classifier. The 13th International Conference on Machine Learning.

Duda, R., Hart, P. (1973). *Pattern classification and scene analysis*. New York: John Wiley & Sons.

El Hindi, K. (2013). Specific-class distance metrics for nominal attributes. In *AI Communications*, *26*, 261–279. IOS Press.

Kammler, D. (2000). *A first course in fourier analysis*. New Jersey: Prentice Hall.

Kavallieratou, E., Sgarbas, K., Fakotakis, N., & Kokkinakis, G. (2003). Handwritten word recognition based on structural characteristics and lexical support. IEEE, Prec. of the Seventh International Conference on Document Analysis and Recognition.

Mahmoud, S.A., & Awaida, S.M. (2009). Recognition of off-line handwritten Arabic (Indian) numerals using multi-scale features and support vector machines vs. Hidden Markov Models. *Arabian J. Sci. Eng., 34*, 429–444.

Mitchell, T.M. (1997). *Machine learning*. New York: McGraw-Hill Science.

Moharir, P.S. (1992). *Pattern recognition transforms*. UK: Research Studies Press Ltd.

O'Hair, Mark.A. (1990). A whole word and number reading machine based on 2-dimensional low frequency fourier transofrms. PhD dissertation, School of Engineering, Air Force Iinstitute of Technology (AU), Wright-Patterson AFB OH.

Riedmiller, M., & Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The RPROP algorithm. Proceedings of the IEEE International Conference on Neural Networks, (pp. 586–591). San Francisco.

Rumelhart, D.E., & McClelland, J.L. (1986). *Parallel distributed processing: Exploration in the microstructure of cognition*. MIT Press Cambridge, MA, USA.

Schaffer, C. (1994). A conservation law for generalization performance. The Eleventh International Conference on Machine Learning. Morgan Kaufmman.

Shartle Gary, F. (1993). *Handwritten word recognition based on fourier co-effcients*. MSc dissertation, School of Engineering, Air Force Iinstitute of Technology (AU), Wright-Patterson AFB OH.

Stanfill, C., & Waltz, D. (1986). Toward memory-based reasoning. 29:12. *Communications of the ACM, 29*, 1213–1228.

Wilson, D.R., & Martinez, T.R. (1997). Improved Heterogeneous distance functions. *Journal of Artificial Intelligence Research, 6*, 1–34.

Zaghloul, R., Bader, D., & Al, Rawashdeh.F. (2012). Recognition of hindi (Arabic) handwritten numerals. *American Journal of Engineering and Applied Sciences, 5*, 132–135.