Handwritten Digit Recognition under Constrained Training Conditions

M. Helali, A. Alneghaimish, I. Ahmad

King Fahd University of Petroleum and Minerals, Saudi Arabia.
{s201265040, s201262360, irfanics}@kfupm.edu.sa

Keywords: Handwritten digit recognition, Arabic digits, training data constraints, artificial data generation.

Abstract

A fundamental step in handwritten digit recognition is to train a system using a large number of handwritten samples. These samples need to be accurately collected and labelled, which is a cumbersome task. In this work, we present several approaches to handwritten digit recognition in situations where little or no handwritten training data is available. We firstly study the effect of the number of training samples per digit on the recognition accuracy. We then study the effect of using machine printed digits in various font typefaces as training data on the performance of the digit recognizer. We then use some image distortion techniques to artificially generate more training data from the machine printed digits. Our final approach is to use the test set for system retraining with the classifier's transcriptions as labels. The results of our system using no labelled handwritten training data are comparable to systems using large handwritten training sets on a benchmark database of handwritten Arabic digits.

1 Introduction and related work

Handwritten digit recognition is an important task with various applications such as postal code recognition, tax form processing and bank check analysis (cf.,[1]). To accurately recognize the digits, a recognition system is trained using a large number of handwritten digits. Such training data needs to be collected and labelled accurately, which is a demanding and time-consuming task, especially if there are no databases available to use (c.f., e.g., [2]). Moreover, training data collected under one environment may not be suitable for training systems to be used under a different environment.

In this paper, we present approaches to handwritten digit recognition when little or no handwritten training data is available. To initialize a system in situations where no handwritten training data is available, we use machine printed digits as the training data. We investigate if promising results can be obtained with the presented techniques as it would reduce, if not completely eliminate, the need for handwritten training data.

Several approaches have been presented in the literature to deal with situation where relatively little handwritten training data is available. Frinken et al. [3] investigated the idea of using co-training to classify handwritten English text, basing their work on the assumption that collecting a large amount of unlabelled training samples is relatively easy compared to labelling them. In co-training, they used two classifiers: HMMs and Long Short-Term Memory Neural Networks. The input for both classifiers is unlabelled samples and the output of each classifier was fed to the other, after filtering out the samples classified with less than a specified confidence threshold. They achieved a statistically significant increase in performance compared to a reference system on the same dataset.

Richarz et al. [4] proposed two semi-supervised multi-view learning strategies for character recognition with relatively few manual annotations. Their goal was to enable deriving a large number of labels using fewer previously labelled data. One of the two approaches they proposed was cluster-level annotations followed by a majority voting to determine unreliable samples. The second approach was pool-based active learning and interactive retrieval with an automatic selection rule to annotate the data. They were able to obtain 92.18% and 95.64% accuracies with the first and second approaches, respectively.

Kozielski et al. [5] presented a method for training a handwritten recognition system using an unsupervised learning mechanism with unlabelled data. They iteratively retrained the system in several rounds, feeding the best hypothesis achieved in each round to the next one. Their system was based on the standard HMM recognizer with features extracted using a sliding window. The recognition system was fed with isolated words after applying pre-processing to the training data. Their unsupervised learning system learned 80% of their training set, which is a promising result.

Other works in the literature investigate the idea of artificial data generation (e.g., [6], [7]). Elarian et al. [6] used isolated Arabic characters and a special connector model to join these characters to form words which can then be used as training data. Varga and Bunke [7] synthesized data to expand the training set. Text lines were synthesized from handwritten text lines by performing a number of geometrical distortions in addition to thinning and thickening of strokes. Miyao and Maruyama [8] proposed a method to improve offline handwritten character recognition by generating artificial data from an online character database. They generated the data using Principal Component Analysis and applied affine transformations to each stroke of both the original characters and the generated ones. They tested their method using an SVM classifier trained on the Japanese Hiragana characters and obtained promising results.

The present work is based on the work of Ahmad and Fink [9]. The authors presented handwritten Arabic text recognition without the use of handwritten training data. The main idea was to use machine printed texts as training data to bootstrap the recognition system. Texts in eight different fonts were used as training data to initialize an HMM-based text recognition system. In addition, unsupervised HMM-adaptation was employed to further improve the results. Finally, the recognition hypothesis on the test set was used to retrain the system. The results of the system are comparable to a text recognition system using labelled handwritten training data on a word recognition task.

The present work differs from the work of Ahmad and Fink [9] in several aspects. Firstly, isolated handwritten digit recognition is performed in the present task as opposed to handwritten word recognition. Isolated digit recognition, although relatively easier as compared to a word recognition task, has different challenges when it comes to using machine printed digits for training the recognition system. As Arabic is inherently cursive both in machine printed and handwritten forms, the similarity of machine printed texts to handwritten texts is higher compared to machine printed digits and handwritten digits. Thus, we investigate whether this approach can work for handwritten digit recognition tasks. Second, we also investigate the impact of artificial generation of more training data by employing distortion techniques such as shear and rotation transformations on machine printed digits. Third, we study the impact of training set size (in terms of number of handwritten training samples per class) on the digit recognition performance.

The rest of the paper is organized as follows: in Section 2, we present our techniques to recognize Arabic digits without handwritten training samples. Next in Section 3, we present our experiment setup, results and discussions. We finally conclude in Section 4 and suggest future possible improvements to the area.

2 Handwritten digit recognition under constrained training conditions

2.1 Training with limited handwritten digit samples

In this step, the effect of the number of handwritten training samples per digit on the classification accuracy is investigated. We want to know how the classifier performs with limited number of handwritten digits and see at which limit we can obtain acceptable results. In addition, we set a performance reference for training using machine printed digits in the next sections. The experiment is conducted in several iterations: in the first iteration, only one sample per class is used, i.e. 10 samples in total. In each following iteration, the number of samples per class is doubled until the maximum number of samples per class is reached, depending on the dataset size in hand.

2.2 Using machine printed digits in one font

One of the main goals of this work is to investigate if recognition of handwritten digits can be achieved using machine printed digits as training data instead of using handwritten digits, since those are hard to collect and annotate. In this step, the classifier is trained using only machine printed digits. The data is generated using different fonts and the classifier is trained on each font separately. Therefore, for each experiment, 10 samples are used for training. The results are compared to the first experiment in the baseline system introduced in the previous section, where the training set has the same size. Since the fonts used have different calligraphic styles, it is expected that the classification accuracy will vary between fonts. We are interested to know which fonts perform better than others and how fonts with complex calligraphic styles compare to those with simple styles.

2.3 Using machine printed digits in multiple fonts

In this step, the machine printed digits from all the fonts are combined to make one training set. We were interested to investigate whether combining the fonts will help increasing the performance, rather than creating noise for the classifier. The results are compared to those of the systems trained on single fonts from the previous section.

2.4 Artificial training data generation by applying image-distortion techniques on machine printed digit images

In this step, the impact of using artificially generated data with image-distortion techniques on the performance of our digit recognizer is investigated. The machine printed digits generated from the previous steps are transformed and added as additional training data. The approach was based on the understanding that the variability of handwritten data is high as compared to machine printed digits. By employing image-distortion techniques we may be able to account for some of the variability during training the recognition system. Three image-transformation techniques have been used: rotation, horizontal shearing and vertical shearing. In rotation, each digit is rotated around the image centre in both clockwise and counter clockwise directions with a constant step size. In horizontal shearing, the width of each image is shrunk and stretched with a constant factor while keeping the height fixed. In vertical shearing, the height of each image is shrunk and stretched with a constant factor while keeping the width fixed. When transforming the digits, care should be taken when choosing the limits and the step sizes. If the generated digits are too dissimilar in shape to the original digit, they will add noise to the classifier and degrade its performance. If they are too similar, on the other hand, the performance of the classifier will not improve much.

2.5 Using test samples' annotations to retrain the classifier

In a real-world scenario, handwritten digits are fed to the system to be classified. The system can use the classification hypotheses of those test samples as labels to retrain the classifier. Provided that the hypotheses are reliable to some extent, the performance of the classifier is expected to increase. In this step, the classifier is retrained using test samples along with the hypotheses generated by the system trained using the training set from the previous step. The classifier is then tested on the same test set and the performance is compared to that of

the previous steps. Clearly for this to work, the test hypotheses should be as close to the true digit classes as much as possible. In the present work, the test samples that are likely to have wrong labels are removed based on the classification confidence. When annotating a test sample, the classifier produces a classification confidence for each class and annotates the sample with the class having the highest confidence. In this work, the test samples are sorted in a descending order based on the maximum classification confidence and the top samples are selected for retraining up to a certain percentage threshold.

3 Experiments and Results

3.1 Dataset

The handwritten digits used for experimentation were obtained from the CENPARMI Arabic digits database, which was introduced in [10] and is widely used in the literature. The handwritten digits are extracted from real bank checks written by many writers. Therefore, the results obtained on the database are more likely to represent the results in a real-world scenario. For more about how the digits are isolated, pre-processed and tagged, readers are referred to [10].

The dataset consists of 10425 digits, separated into two sets: 70% training and 30% testing. We further split the training set randomly into training and development sets. Therefore, our training-development-test configuration is: 60% training, 10% development and 30% testing sets. Parameter selection and calibration were done based on the results on the development set while the system was evaluated on the test set.

3.2 Classifier

For the classifier, we used a Random Forest [11] classifier. Random forest is an ensemble method that uses decision trees as weak classifiers. Each tree is initialized using a random set of features from the whole feature set. Each tree annotates the sample with a class and the final decision is made using majority voting of all decision trees.

In our experiments, the number of trees (weak learners) is 200, the maximum depth of a tree is 50, the minimum leaf size is 1 and the random number generator seed is 1.

3.3 Features and Pre-processing

We adapted the features presented in Wienecke et al. [12] for the digit recognition task. Nine geometrical features are computed from stripes of images using the sliding window technique running across the digit images. The window width is 8 pixels with an overlap of 4 pixels, i.e., a shift of 4 pixels. The bottom of the image is regarded as the baseline for feature computation. The computed features are listed in Figure 1. The digit images are normalized by setting the image width to 64 pixels and keeping the aspect ratio unchanged.

1. The average distance from the bottom of the image to the upper contour of the ink pixels.
2. The average distance from the bottom of the image to the lower contour of the ink pixels.
3. The average distance from the bottom of the image to the center of gravity of the ink pixels.
4. The angle of the upper contour of the ink pixels with respect to the horizontal axis.
5. The angle of the lower contour of the ink pixels with respect to the horizontal axis.
6. The angle of the center of gravity of the ink pixels with respect to the horizontal axis.
7. The average of the number of black-to-white transitions per column.
8. The percentage of ink pixels in a frame.
9. The average number of ink pixels between the upper and lower contours of the ink pixels.

Figure 1: The list of features extracted from the digit images (adapted from [12]).

3.4 Training with complete training set of handwritten digits

As a first set of experiments, we train the classifier using the training set of handwritten digits from the CENPARMI database [10]. We use a total of 6275 digit samples for training. The parameters were calibrated based on the system's recognition performance on a separate development set containing 1113 digit samples which were randomly taken out from the official training set. Finally, the system was evaluated on the official test set of the database containing 3035 digit samples. This experiment was carried out in order to set a baseline result for the experiments in the following sections. In addition, the results are used to compare our system with the state-of-the-art on the same dataset. We report the results in terms of *error rate* (%), i.e., the number of misclassified digits over the total number of digits classified, in percentage. Table 1 shows the recognition results along with the state-of-the-art results on the same dataset. It should be noted that, based on our understanding, other works have not optimized the system parameters using a separate development set but evaluated the systems directly on the test set. We see from the results that our recognition system is comparable to the state-of-the-art on the same database. The confusion matrix is presented in Table 2. From the confusion matrix we can see that most of the error are between digits 0 and 1.

Work	Classifier	Error Rate
Assayony and Mahmoud [13]	SVM	0.66
Mahmoud and Al-Khatib [14]	SVM	1.05
Alamri et al. [1]	SVM	1.52
Giménez et al. [15]	Bernoulli Mixture Models	2.0
Present work	Random Forest	1.45

Table 1: A comparison between the error rate of the present work and the state-of-the-art on the CENPARMI dataset.

Actual digits	Predicted digits									
	0	1	2	3	4	5	6	7	8	9
0	1563	10	0	0	0	1	0	0	0	0
1	7	296	0	0	0	0	1	0	0	0
2	1	0	223	0	1	1	0	0	0	0
3	0	0	2	141	1	0	0	0	1	0
4	1	1	2	0	129	0	0	0	0	0
5	1	0	0	0	0	260	0	1	0	0
6	0	1	0	0	0	0	110	0	0	0
7	2	0	0	2	0	0	0	105	0	0
8	4	0	0	0	0	1	0	0	92	0
9	1	1	0	0	0	0	0	0	0	72

Table 2: Confusion matrix for digit recognition on the test set of the CENPARMI database.

3.5 Training with limited handwritten digits

In this experiment, we limit the number of handwritten samples per digit used in training. We train using random samples from the training set of the database. In the first step, we use only one sample per digit; in each following step, we double the number of samples per digit until we reach the maximum limit per digit, which is 256 because some digits have more samples than others. The results are presented in Figure 2.

It is clear from the results that the recognition accuracy increases as the number of training samples increases. We also note that 2.7% error rate can be obtained using 64 samples per digit, i.e. 640 samples in total, which is an acceptable result compared to our baseline result in the previous section.

3.6 Using machine printed digits in one font

In this step, we use only machine printed digits for training. We selected 8 Arabic fonts with different calligraphic styles in order to cover the different variations in writing styles of handwritten digits. Figure 3 shows sample images of digits 0-9 (from left to right) using the 8 different fonts along with samples of handwritten digits (top row).

The system is trained using digits of each font separately and the error rate is observed; the results are shown in Table 3. The first observation from the results is that relatively low error rates can be obtained using single fonts, i.e. only 10 training samples. Second, by comparing this result to the first run in the previous experiment, we see that the error rate for some fonts is comparable to that when using handwritten digits, i.e., one handwritten image per digit. We also note that *Ahram* font's performance was superior despite its simple style, indicating that it is not a necessary condition for low error rates to use a font with complex style.

3.7 Using machine printed digits in multiple fonts

In this step, we combine the digits from all fonts to make a larger and more diverse training set. The result of this experiment is shown in Table 4. From the result, we note that combining the fonts does increase the accuracy rather than adding noise. This is an interesting find because it agrees with the results in the previous work on Arabic text [9], where a significant increase of accuracy was achieved by adding more fonts. It is also possible that using additional fonts will further decrease the error rate.

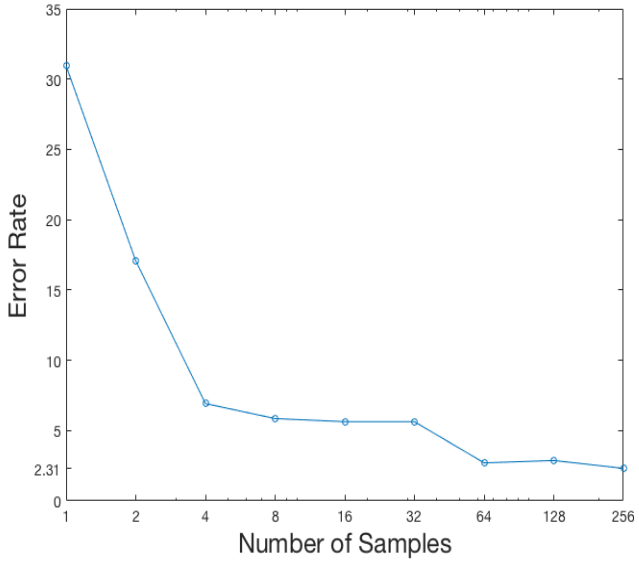

Figure 2: Error rate of training using different numbers of samples per digit.

Font name	Error Rate
Abasan	68.10
Ahram	**31.90**
Basamat	49.78
Damas	60.83
Neqat	44.56
Roqaa	71.70
Sindibad	49.06
Text	75.11

Table 3: Error rate using different fonts for training.

Figure 3: Arabic (Indian) handwritten digits (top row) from the CENPARMI database along with sample machine-printed digits in 8 fonts used in the experiments.

3.8 Artificial generation of training data

In this step, we use the set of all fonts combined obtained from the previous step and apply three image-transformation techniques: rotation, horizontal shearing and vertical shearing. The algorithm for the artificial generation of training data using the above image-transformation techniques is presented in Figure 4.

Figure 5 shows two sample font digits on the left and example results after applying rotation, horizontal shearing and vertical shearing transformations, respectively. For every font, the number of samples per digit after applying the transformations is 30, plus one sample representing the original digit. Therefore, the total number of samples in the training set is 2480 samples. The result of using this training set is presented in Table 5.

We observe from the result that applying image transformations decreases the error rate by almost a half from the previous result. This was achieved using only three image-transformation techniques. It is possible that implementing more techniques can further improve the performance. For the rest of experiments in this work, we use the training set of all fonts with transformations, since it gives the best result so far.

Fonts	Error Rate
All fonts together	23.76

Table 4: Error rate of training using all fonts combined.

1. For θ, where $-10 \leq \theta \leq 10$, with step size of 2, rotate the image for θ degrees.
2. For i, where $0.5 \leq i \leq 1.5$, with step size of 0.1, scale the height of the image by i while maintaining the width.
3. For j, where $0.5 \leq j \leq 1.5$, with step size of 0.1, scale the width of the image by j while maintain the same height.

Figure 4: Algorithm for artificial training data generation.

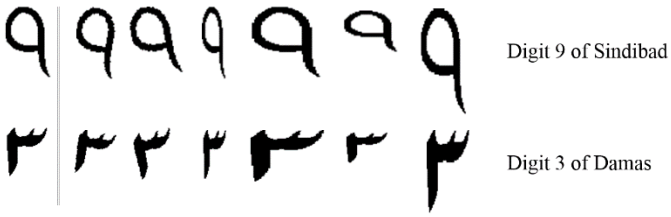

Figure 5: Example results of applying image-distortion techniques on two font samples.

Training Set	Error Rate
All fonts + transformations	12.59

Table 5: Error rate of using multiple fonts with transformations.

3.9 Using test samples' annotations to retrain the classifier

In this step, we use the trained classifier from the previous steps to annotate test samples and then feed those samples with their annotations as training data for the classifier. The annotations should be reliable enough in order to achieve high recognition rates. As discussed earlier, we sort the test samples based on the classification confidence and select the top samples up to a certain percentage threshold. In the first part of the experiment, we tested several values of the threshold using the training set of all fonts and transformations; the results are shown in Figure 6. We see that the least error rate is obtained when retraining using the top 85% test samples.

In the second part of the experiment, we use the training set of all fonts and transformations to annotate the test set and select the top 85% samples for retraining. We then feed them to the classifier and test it on the test set. We do this procedure for multiple iterations until the error rate converges. The results are shown in Table 6. It is clear from the results that there is a decrease in the error rate from the first iteration due to the use of handwritten digits for training. The error rate keeps decreasing and plateaus in iteration 4, before starting to increase in subsequent iterations. The results from this step are comparable to the state-of-the-art on the same benchmark dataset as presented in Table 1. This means that all previous steps can be regarded as an initialization step to confidently annotate the test samples and use them to train the classifier. It is to be noted that this step does not conflict with real-world settings because the system will be tested on handwritten digits anyways.

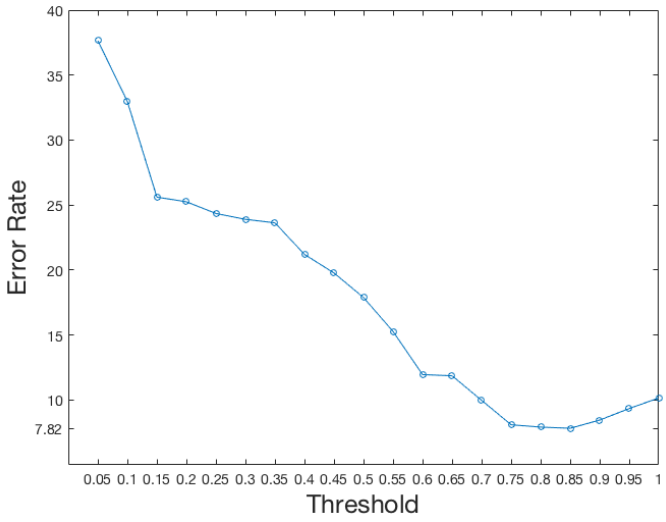

Figure 6: Error rate of testing different threshold values for selecting top test samples.

Iteration	Error Rate
1	7.81
2	5.5
3	3.56
4	**2.96**
5	3.03
6	3.26

Table 6: Error rate of retraining using test set annotations as labels.

4 Conclusions

In this paper, we investigated handwritten digit recognition when little or no handwritten training data is available. We firstly tested the effect of the number of training samples per digit on the classification accuracy. We then used machine printed digits to train the classifier on each font separately and obtained significantly high recognition rates. Next, we combined all fonts in a single training set and achieved an increase in the recognition accuracy. Then, we generated artificial training samples by applying three image-transformation techniques on font digits. The results show an increase in the recognition accuracy. Finally, we used the training set from previous steps to initialize a classifier and used its test samples' annotations as labels and fed back the test samples as training data. We achieved further increase in the recognition accuracy. The results are comparable to those achieved on the same benchmark dataset using a large number of handwritten digits.

Future works in the area can explore the different choices of fonts and font combinations: since the recognition accuracy of each font is different, using a subset of the fonts or a different font set can yield better results. Another area to explore is the choice of the image-transformation techniques to generate artificial training data: the present work uses only three transformation techniques. Using different transformation techniques instead of/besides the present ones can also ameliorate the results.

Acknowledgements

The authors would like to thank King Fahd University of Petroleum and Minerals (KFUPM) for supporting this research under the undergraduate research grant no. URGS1601.

References

[1] H. Alamri, C. He, and C. Y. Suen, "A New Approach for Segmentation and Recognition of Arabic Handwritten Touching Numeral Pairs," *Comput. Anal. Images Patterns*, vol. 5702, pp. 165–172, 2009.

[2] S. A. Mahmoud, "Recognition of writer-independent off-line handwritten Arabic (Indian) numerals using hidden Markov models," *Signal Processing*, vol. 88, no. 4, pp. 844–857, 2008.

[3] V. Frinken, A. Fischer, H. Bunke, and A. Foornes, "Co-training for Handwritten Word Recognition," in *Proceedings of the Eleventh International Conference on Document Analysis and Recognition (ICDAR 2011)*, 2011, pp. 314–318.

[4] J. Richarz, S. Vajda, and G. A. Fink, "Annotating handwritten characters with minimal human involvement in a semi-supervised learning strategy," in *2012 International Conference on Frontiers in Handwriting Recognition*, 2012, pp. 23–28.

[5] M. Kozielski, M. Nuhn, P. Doetsch, and H. Ney, "Towards Unsupervised Learning for Handwriting Recognition," in *Proceedings of the 14th International Conference on Frontiers in Handwriting Recognition (ICFHR 2014)*, 2014, pp. 549–554.

[6] Y. S. Elarian, I. Ahmad, S. M. Awaida, W. G. Al-Khatib, and A. Zidouri, "An Arabic handwriting synthesis system," *Pattern Recognit.*, vol. 48, no. 3, pp. 849–861, Mar. 2015.

[7] T. Varga and H. Bunke, "Perturbation Models for Generating Synthetic Training Data in Handwriting Recognition," Springer Berlin Heidelberg, 2008, pp. 333–360.

[8] H. Miyao and M. Minoru, "Virtual Example Synthesis Based on PCA for Off-Line Handwritten Character Recognition," in *Document Analysis Systems VII*, vol. 3872, H. Bunke and A. L. Spitz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 96–105.

[9] I. Ahmad and G. A. Fink, "Training an Arabic handwriting recognizer without a handwritten training data set," in *Proceedings of the 13th International Conference on Document Analysis and Recognition (ICDAR 2015)*, 2015, pp. 476–480.

[10] Y. Al-Ohali, M. Cheriet, and C. Y. Suen, "Databases for recognition of handwritten Arabic cheques," *Pattern Recognit.*, vol. 36, no. 1, pp. 111–121, Jan. 2003.

[11] Tin Kam Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, 1998.

[12] M. Wienecke, G. A. Fink, and G. Sagerer, "Toward automatic video-based whiteboard reading," *Int. J. Doc. Anal. Recognit.*, vol. 7, no. 2, pp. 188–200, 2005.

[13] M. Assayony and S. Mahmoud, "An Enhanced Bag-of-Features Framework for Arabic Handwritten Sub-words and Digits Recognition," *J. Pattern Recognit. Intell. Syst.*, vol. 4, no. 1, pp. 27–38, 2016.

[14] S. A. Mahmoud and W. G. Al-Khatib, "Recognition of Arabic (Indian) bank check digits using log-gabor filters," *Appl. Intell.*, vol. 35, no. 3, pp. 445–456, May 2010.

[15] A. Giménez, J. Andrés-Ferrer, A. Juan, and N. Serrano, "Discriminative Bernoulli mixture models for handwritten digit recognition," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2011, pp. 558–562.