# Efficient Gabor-Based Recognition for Handwritten Arabic-Indic Digits

Emad Sami Jaha
Department of Computer Science
King Abdulaziz University, Jeddah 21589, KSA

***Abstract*—In daily life, the need of automatically digitizing paper documentations and recognizing textual images is still present with existing and potential upcoming rooms for improvements, especially for languages like Arabic, which is unlike English as an instance, has more complex context and not been extensively supported by research in a such domain. As yet, the available online offline optical character recognition (OCR) systems have utilized functional techniques and achieved high performance mainly on machine printed data images. However, in case of handwritten script, the recognition task becomes highly unconstrained and much more challenging. Amongst a large verity of recognizable multi-lingual characters, handwritten digit recognition is a considerably useful task for different purposes and countless applications. In this research, the focus is on Arabic (known today as Indic or Indian) digit recognition using different proposed Gabor-based approaches in several combinations with different classification methods. The proposed approaches are trained and tested using 91120 digit samples of two independent standard databases (Arabic-Handwritten-Digits and AHDBase), allowing performance variability assessments and comparisons not only between the different combinations of features and classifiers but also between different datasets. The proposed Arabic-Indic digit recognition system achieves high recognition rates reach up to 99.87%. This research practically shows that one of the proposed approaches with significant dimensionality reduced features remains attaining a high recognition rate with low complexity time, which can be hence recommended further for online digit recognition systems.**

***Keywords*—Digit recognition; Gabor filters; OCR; k-nearest neighbor; artificial neural networks**

## I. Introduction

Continuous advances in the realm of hardware and software have paved the way for numerous nowadays impressive and effective technologies. Optical character recognition (OCR) is one apparent instance of such technologies that has been deployed for the use in several online/offline systems and in different operational forms [1], [2]. Even handheld devices like smartphones and tablets have emerged as new platforms for plenty functional OCR apps exploiting their built-in cameras for scanning and other hardware for required processing and operation, so they can be usable in almost any ambient environments. Note that, while text recognition in different languages is considered within the most challenging pattern recognition problems, analyzing unconstrained handwritten scripts is deemed as much more difficult [3] and challenging recognition task [4], [5]. This is owing to the vast variability producing a lot of visual differences in terms of shape, size, slant, pen stroke and deformation [5], mostly when applied to languages with complex writing characteristics and rules [6].

Although, majority of existing technologies and research efforts in the literature have been devoted intensively in supporting widely used languages such as English in either character or digit recognition, other languages with more complex context and fully cursive writing style like Arabic has not yet received adequate technical advances and research investigations. As many research studies were undertaken for English in either general handwritten character recognition [2] besides typewritten [3] or limited to only digit recognition in both machine-printed [6], [7] and human handwriting [4], [8]-[14] forms. Thus, over the times, more research interests are increasingly attracted to the scope of handwritten Arabic OCR [5], [6], [15]-[17].

Particularly, the focus in this research is on recognizing handwritten Arabic digits known today also as Indic or Indian digits. This recognition capability is very useful and can be utilized in many applications in a variety of systems and purposes. Therefore, this research field has recently received some research interests [16], [18]-[20]. For the sake of creating databases of Arabic-Indic digit samples and providing them for further research work, a number of research works have produced such databases and introduced them with initial studies/results to be used as benchmarks for future studies [15], [21], [22].

Dozens of viable approaches can be developed and used for character and digit recognition purposes in different languages [1], [4], [23]. A number of well-known techniques, which were introduced for other machine learning and pattern recognition applications, may also be used in a diversity of character and digit recognition contexts [1], [2], [24]. Artificial Neural Network (ANN) as an example, has been excessively applied in different topologies for learning to recognize and classify characters and numerals [23], such as Multilayer [7], [18], [21], Probabilistic [3], Convolutional [6], [19], [25], and Back Propagation [9], [10], [14], [20] Neural Networks. Moreover, other common classification methods were employed and found useful for attaining high recognition rates such as single- [13], [22] or multi- Nearest Neighbor [18], Support Vector Machine (SVM) [11], [13], [22], Deep Learning [23], [25], and Random Oblique Decision Trees [12]. Many related experimental studies were carried out via frequently used feature extraction methods including Gabor filters [7], [13], and SIFT and GIST descriptors [12]. Further applicable feature extraction methodologies were also involved in this domain such as vertical or horizontal

histograms [11], and different basic dimensional, spatial, or directional aspects of a letter or digit sample [18].

The main contributions of this research are:

- a proposed system for handwritten Arabic-Indic digit recognition in offline mode with a recommendation for a potential online counterpart for future;
- different proposed Gabor-based approaches using mainly two algorithmic methods for feature selection and dimensionality reduction;
- an extended analysis from different aspects for performance assessment and comparison between different combinations of features and classifiers applied on two independent standard large-scale databases; and
- an investigation for the effects of using the same proposed approaches under the change of database, enabling performance variability evaluation, validation, and comparison.

Noteworthy, throughout this research paper we use the phrase of "Arabic digit" by all means to refer to a digit writing style also known as Indian- or Indic- digit, is the commonly used style by Arabs nowadays, as shown in Table 1 along with the corresponding worldwide style for each digit.

In the rest of this paper, Section 2 provides brief introduction about the concepts of Gabor filters and technically describes a number of created and used Gabor filters. Section 3 describes the databases of Arabic-Indic digit images used for training and validating the proposed system. Detailed feature extraction methods for both the major Gabor based and the additional supplementary features are illustrated in Section 4. The adopted classifiers are presented in Section 5. Section 6 explains the methodology of the conducted experimental work and analyzes the obtained results from different perspectives. Finally, the conclusions and brief discussions are given in Section 7.

## II. Gabor Filters

Gabor filters increasingly attract more research interests, due to their latent ability for effective analysis and distinctive feature representation of visual appearance in image and video data. Such Gabor filters have been effectively used in different fields including computer vision, image processing, document analysis, and pattern recognition [13]. Gabor filters have been further extended to the use in biometric applications, such as face detection and recognition, iris recognition, and fingerprint recognition [26]. Numerous existing systems and applications confirm the potency and usefulness of Gabor-based analysis and features in different forms [25]. Object tracking, texture analysis, and multilingual optical character or digit recognition are present various applications, where Gabor filters have been employed [27].

The complex 2D Gabor filter can be mathematically described in different formulations, where one possible formula can be defined in the spatial domain [28] as follows:

TABLE I. Handwritten and Printed Arabic Digit Style with their Corresponding Worldwide Style for Each Digit

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Arabic (Indic)** | Handwritten | ٠ | ١ | ٢ | ٣ | ٤ | ٥ | ٦ | ٧ | ٨ | ٩ |
| | Printed | ٠ | ١ | ٢ | ٣ | ٤ | ٥ | ٦ | ٧ | ٧ | ٩ |
| **Worldwide** | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

$$\Psi_S(x,y;f,\theta)=\frac{f^2}{\pi\gamma\eta}\,e^{-\left(\frac{f^2}{\gamma^2}x'^2+\frac{f^2}{\eta^2}y'^2\right)}\,e^{j2\pi f x'} \tag{1}$$

Where

$$x'=x\cos\theta+y\sin\theta,\quad y'=-x\sin\theta+y\cos\theta$$

Here $x$ and $y$ are the spatial coordinates of the filter. $f$ denotes the central frequency parameter. The rotation angle of the Gaussian major axis and the plane wave is symbolized by $\theta$. $\gamma$ and $\eta$ are parameters used to control the bandwidth, where $\gamma$ and $\eta$ are the sharpness along the major and minor axes respectively.

This 2D Gabor filter can be also represented in the frequency domain as follows:

$$\Psi_F(u,v;f,\theta)=e^{-\frac{\pi^2}{f^2}(\gamma^2(u'-f)^2+\eta^2v'^2)} \tag{2}$$

Where

$$u'=u\cos\theta+v\sin\theta,\quad v'=-u\sin\theta+v\cos\theta$$

In the frequency domain the $x$ and $y$ coordinates are replaced by $u$ and $v$, which are the frequency variable pair of the filter. For further practical implementation in the frequency domain, as defined in (3), the Fast Fourier Transform (**FFT**) is applied to a digit sample image $I(x, y)$, then the output transformed image is multiplied by the Gabor filter in the frequency domain defined by (2), as such, the convolution of the Gabor filter and the Fourier transformed image is performed. Eventually the resulting response, which represents the initially extracted Gabor-based features, is computed by applying the Inverse Fast Fourier Transform (**IFFT**) to the output of the convolution operation, as in (3).

$$g(u,v;f,\theta)=\mathbf{IFFT}\left[\Psi_F(u,v;f,\theta)\otimes\mathbf{FFT}\left[I(x,y)\right]\right] \tag{3}$$

In this research 24 different Gabor filters with different four scales (0, 1, 2, and 3) and six orientations (0°, 30°, 60°, 90°, 120°, and 150°) are generated, as demonstrated in the spatial domain in Fig. 1, where each orientation is $\theta_k$ is computed by $\theta_k=k\pi/n$, where $k=\{0, 1\ldots, n-1\}$ and n=6 the number of used orientations.

Fig. 2 shows a normalized sample image of digit '9', the transformed version of the same image via **FFT**, and the convolution of the transformed image with each of the generated 24 Gabor filters represented in the spatial domain.

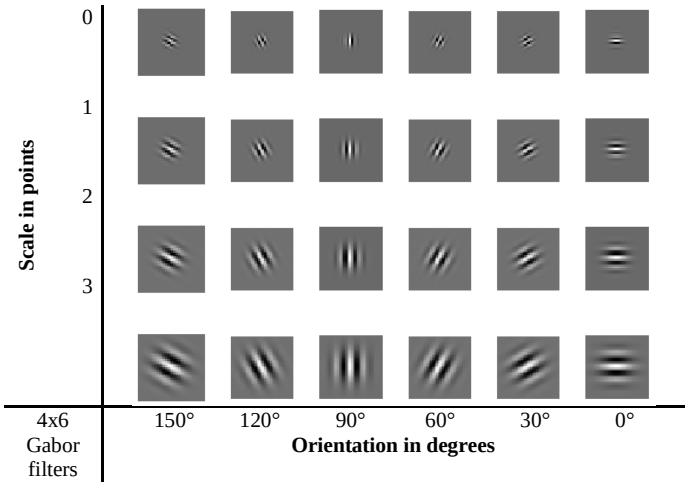



Fig. 1. The 24 Created and used Gabor Filters.

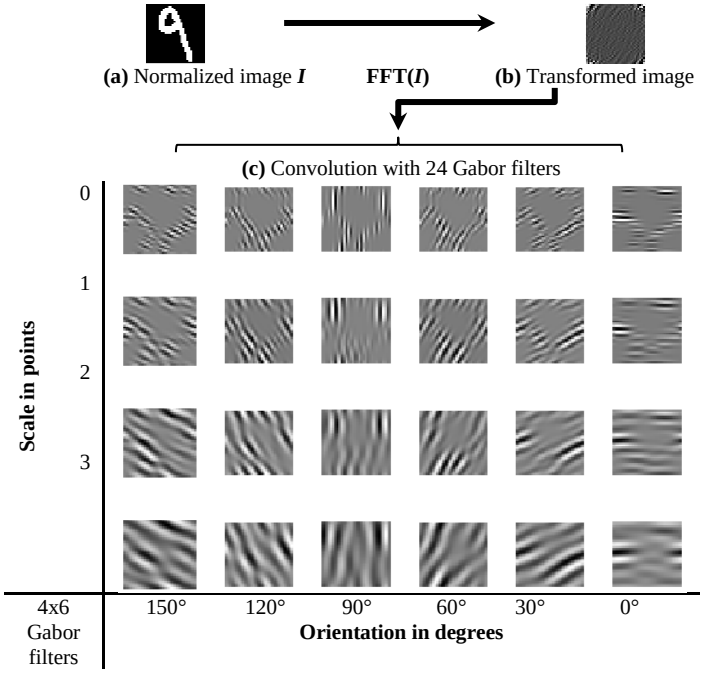



Fig. 2. (a) Normalized Sample Image of Digit 9, (b) The Transformed Image Via FFT, and (c) The Convolution for Each of the 24 Gabor Filters with the Transformed Image.

## III. Arabic-Indic Digit Databases

In this research, two different databases are used for conducting the experimental work, since they are publicly available standard databases. This enables performance variability evaluation, validation, and comparison. Noting that, both databases comprise handwritten Arabic digit samples from zero to nine collected from a number of writers and they are designated for Arabic-Indic digit classification purposes.

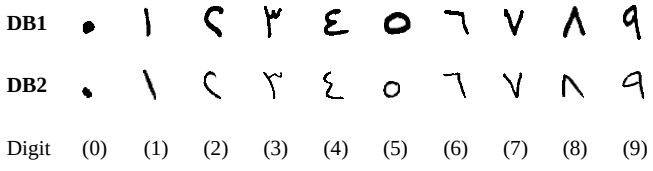



Fig. 3. Digit Samples (0-9) from the used Databases.

TABLE II. Descriptions of the used Databases

| Characteristic | DB1 | DB2 |
|---|---|---|
| Number of writers | 44 | 700 |
| Total samples per digit (0-9) for each writer | 48 | 10 |
| Total digit samples per writer | 480 | 100 |
| Total samples per digit (0-9) | 2112 | 7000 |
| Total digit samples | 21120 | 70000 |

TABLE III. Training and Testing Datasets Per Database

| Database | DB1 | | DB2 | |
|---|---|---|---|---|
| **Dataset** | Training | Testing | Training | Testing |
| **Total digit samples** | 14960 | 6160 | 60000 | 10000 |
| **% of database** | 70% | 30% | 85.71% | 14.29% |
| **Total samples per digit (0-9)** | 1496 | 616 | 6000 | 1000 |

The first database of Arabic handwritten digits [22] consists of 21120 samples and will be referred to in this research as DB1. The second is a larger database known as AHDBase [21] comprising a total of 70000 samples and will be referred to in this research as DB2. Table 2 shows brief descriptions for both DB1 and DB2 databases used in this research. Fig. 3 illustrates some random digit samples of all ten digits in Arabic-Indic format (i.e. '٠', '١', '٢', '٣', '٤', '٥', '٦', '٧', '٨', '٩') from each database, representing respectively the digits from '0' to '9'.

For experimental work, each database was partitioned into two portions training and testing datasets. The training dataset is a larger set exclusively used for classifier training purposes while the testing dataset is exclusively used for performance evaluation. Note that training and testing datasets have exclusive data from both writer and sample aspects. Table 3 illustrates the training and testing datasets for DB1 and DB2.

## IV. Feature Extraction

### A. Gabor-based Feature Extraction

As a preprocessing, firstly, a binary digit image is inverted to set the background pixels to zeros and the foreground pixels to ones, if not already so in the database. Secondly, the digit image is simply normalized to 32×32 px in case it has equal width and height. Otherwise the larger dimension (either width or height) is resized to 32 px, whereas the smaller dimension is relatively resized to the original image size. Then padding is applied to the smaller dimension by adding zero pixels from both sides along that dimension (namely expanding the black background) to reach the 32 px, resulting in a 32×32 normalized digit image. Fig. 4 shows few digit samples before and after normalization.

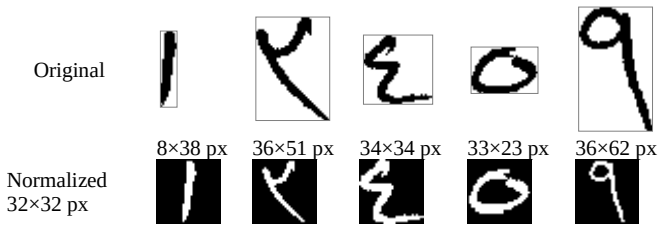



Fig. 4. Original and Normalized Digit Samples.

Thirdly, as feature extractors, each of the created 24 Gabor filters, described in Section 2, is used to apply convolution on the normalized digit image. The resulting 24 filtered digit images (of size 32×32 px each) are concatenated to form one 128×192 (large) feature matrix, such that they are consistently reshaped (organized) as 4×6 filtered digit images (matrices) to eventually compose the large matrix; similar to the order shown in Fig. 1. Finally, all values of the large matrix are mapped to fit between -1 and 1, then two proposed feature selection approaches based on dimensionality reduction are applied each to compose a single Gabor-based feature vector.

*1) Dimensionality reduction*: The first proposed feature selection approach is referred to as App-1 and concerned with applying a dimensionality reduction method to the Gabor-based 128×192 feature matrix, using the following Algorithm-1.

***Algorithm-1***

1. Let $M$ be a Gabor-based 128×192 feature matrix, where $R$=128 and $C$=192 are the total number of rows and columns respectively;
   let $d$=6 be the initial dimensionality reduction level;
   let $i$ be an integer indicates the current group of $d$ rows in $M$, were $1 \leq i \geq |R/d|$; and
   let $j$ be an integer indicates the current group of $d$ columns in $M$, were $1 \leq j \geq |C/d|$
2. Remove the $d^{th}$ row from each $i$ group of $d$ rows from 1 to $|R/d|$
3. Remove the $d^{th}$ column from each $j$ group of $d$ columns from 1 to $|C/d|$
4. If $d \geq 2$, set $d$=$d$-1 and go to Step 2
5. Reshape the resulting 22×32 reduced feature matrix $M'$ as a single 704-value feature vector $v$

*2) Mean and mean-covariance of reduction*: The second proposed feature selection approach is referred to as App-2, which is very similar to App-1 but that it involves further computation to infer the mean and mean-covariance of the reduced Gabor-based feature matrix leading to a smaller number of effective feature vector, by using the following Algorithm-2.

***Algorithm-2***

1. Let $M$ be a Gabor-based 128×192 feature matrix, where $R$=128 and $C$=192 are the total number of rows and columns respectively;
   let $d$=5 be the initial dimensionality reduction level;
   let $i$ be an integer indicates the current group of $d$ rows in $M$, were $1 \leq i \geq |R/d|$; and
   let $j$ be an integer indicates the current group of $d$ columns in $M$, were $1 \leq j \geq |C/d|$
2. Remove the $d^{th}$ row from each $i$ group of $d$ rows from 1 to $|R/d|$
3. Remove the $d^{th}$ column from each $j$ group of $d$ columns from 1 to $|C/d|$
4. If $d \geq 2$, set $d$=$d$-1 and go to Step 2
5. Compute the 39-value mean vector $\mu$ of the 26×39 reduced feature matrix $M'$
6. Compute the 39-value mean vector $s$ of the covariance matrix of the 26×39 reduced feature matrix $M'$
7. Concatenate $\mu$ and $s$ vectors as a single 78-value feature vector $v$

### *B. Supplementary Feature Extraction*

With a view to supplementing the Gabor-based features and improving the capability of scale invariant digit recognition, we use four additional features that are immune to change in scale and may support discrimination between ambiguous handwritten digits based on basic dimensional and spatial aspects of a digit sample. Although similar simple features were proposed and used earlier as major features for digit recognition [18], [21], we utilize four features here as mere supplementary features along with the mainly used Gabor-based features. Moreover, unlike related existing approaches, all adopted four features preserve scale (size) invariance characteristic, where none is extracted by involving any average size or average dimension within all samples of a training dataset.

*1) Ratio of width to height*: As a basic dimensional-based feature, the ratio between the width $w_x$ and height $h_x$ is inferred for the bounding box of an unnormalized digit sample x, which is referred to here as Ftr1 and can be simply defined as:

$$Ftr1_x = w_x / h_x \quad (4)$$

Such a feature, was earlier used as a major feature and highlighted to be a helpful discriminative feature between digits ‘0’ and ‘1’ [21].

*2) Ratio of foreground pixels to all pixels*: For an unnormalized binary digit image x, this feature is calculated by dividing the number of all foreground pixels $|fg_x|$ by the total number of image pixels including both foreground and background pixels, which can be simply calculated by multiplying the with $w_x$ by the height $h_x$. This feature is considered as a basic spatial-based feature and referred to here as Ftr2. It is expected to help differentiating between digits ‘0’ and ‘5’, as signified in [21]. The formulation of this feature can be defined as follows:

$$Ftr2_x = |fg_x| / (w_x * h_x) \quad (5)$$

*3) Ratio of foreground pixels to the square of normalized digit-larger-dimension*: This feature is referred to as Ftr3 and can be extracted from an unnormalized binary digit image by computing the ratio of the number of foreground pixels to the number of pixels in a square shaped area with sides equal to the larger dimension (either width or height) of the normalized digit bounding box (see Section 4.1). This feature assists distinguishing digit '0' from digits '1' and '5', which are mostly confused with it. Thus, Ftr3 extraction can be defined such that.

$$Ftr3_x = |fg_x| / \max(w_{x'}, h_{x'})^2 \quad (6)$$

Where $x$ is the unnormalized binary digit image, whereas $x'$ is the normalized same digit image. $|fg_x|$ is the number of foreground pixels in image $x$. $\max(w_{x'}, h_{x'})^2$ is a function used to decide which dimension of the normalized $x'$ digit bounding box is larger either the width $w_{x'}$ or the height $h_{x'}$ in order to be squared and then used for deriving the required ratio.

*4) Ratio of digit diagonal to normalized digit diagonal*: This feature is derived as a ratio between two different diagonals and referred to as Ftr4. The first diagonal is computed for an unnormalized binary digit image as the summation of the squared width and height of the digit bounding box. Whereas the second diagonal is deduced as two times the squared value of the larger dimension (either width or height) of the normalized digit bounding box. This feature supports differentiation of a randomly shaped digit '0' that could mimic any of the other digits. As such, Ftr4 can be derived using the following formulation.

$$Ftr4_x = (w_x^2 + h_x^2) / [\max(w_{x'}, h_{x'})^2 * 2] \quad (7)$$

Where, $w_x^2$ and $h_x^2$ are the squared width and height of the unnormalized digit image $x$ respectively, which are summed to calculate the first diagonal. While $\max(w_{x'}, h_{x'})^2$ is a function used to find out which dimension of the normalized $x'$ digit bounding box is larger either the width $w_{x'}$ or the height $h_{x'}$ in order to be squared and then multiplied by two to compute the second diagonal and to finally derive the ratio between both diagonals.

## V. Classification

Towards achieving an extended analysis and a comparative study to the viability of the proposed features approaches, we adopt two different classification methods to take place in our Arabic-Indic handwritten digit recognition system.

### A. K-Nearest Neighbor (k-NN)

K-Nearest Neighbor (k-NN) is a commonly used classifier for a wide variety of recognition applications and categorization purposes. It is deemed as a straightforward but also a powerful discriminative classification method [18].

The learning mechanism of *k*-NN can be described as instance-based learning. Namely, it does not involve any prior stressful training phase or function approximation for building a classification model. It rather investigates the likelihood between a test sample and all other training samples represented in the multidimensional feature space. The likelihood is computed as the distance between a test and a training sample (feature vectors) in the feature space, which can be inferred using any distance function such as the most commonly used Euclidean distance.

As such, *k* closest samples (neighbors) are retrieved with their labels and the most frequent label is nominated to be the likely class of that test sample. *k* is typically a small positive integer and is more functional to be an *odd* (not *even*) number to avoid as possible the random class selection, for instance, in case of having two equal numbers of neighbors suggesting two different classes.

In this research we apply *k*-NN classifier via Euclidean distance and use it in two modes with *k*=1 and *k*=3, based on the same *k*-NN implementation method used in [29]. Nevertheless, *k*-NN is used here for classification purpose and to nominate *k* nearest neighbors of a test feature-vector. Thus, the likelihood is estimated as the sum of Euclidean distance between each test feature-vector and all training feature-vectors, resulting in an ordered list of all training digit-samples based on likelihood. The labels/classes of the top *k* digit-samples in that list are nominated to be the potential labels/classes of the test digit-sample, where a repeated label/class emphasizes its likelihood and votes to itself as the candidate class.

### B. Feed-Forward Neural Network (FFNN)

The two-layer feed-forward neural network (FFNN) is a widely used model amongst several neural network topologies [30]. Its mechanism depends on a unidirectional information flow staring form inputs to the output layer. However, it often implicates back-propagation learning rule that propagates the information form the output layer through the hidden layer backward to the input layer [31]. As such, it computes the sensitivity of a cost function with respect to each weight and correct the error by updating each weight proportional to the sensitivity [32].

Although the two-layer FFNN comprises of a simple structure in comparison with other more complex models embracing multiple hidden layers, it is also fast and powerful in solving difficult classification problems. Since it has only one hidden layer, the number of hidden neurons has to be carefully selected to balance between enforcing desirable modeling and avoiding over fitting issue [18].

In this research, the input length is equal to the number of features in the feature vector with respect to the used approach, whereas the number of neurons in the output layer is equal to ten which is the number of classes or digits from 0 to 9. Besides, the number of hidden neurons is chosen to be equal to 300, which are used alongside the output layer to train the adopted neural net using samples of the corresponding training dataset for modeling the Arabic digit classification problem.

## VI. Experiments and Analyses

As introduced in Section 4, two Gabor-based approaches *App-1* and *App-2* are proposed and examined in handwritten

Arabic digit recognition on DB1 and DB2 databases likewise. For each approach, three experiments are conducted per database using the classifiers 1-NN, 3-NN, and FFNN, described in Section 5. Table 4 summarizes the various experiments conducted in this research along with the resulting accuracies. In each of the 12 conducted experiments, all digit images in a training dataset of a designated database are used to train an associated classifier for learning a discriminative classification model. Fig. 5 illustrates an overview of the experimental framework of the proposed Handwritten Arabic-Indic digit recognition system.

Whiles all digit images in a corresponding test dataset from the same database are used accordingly to evaluate the recognition performance of the entire approach (see Table 3). The preprocessing, described in Section 4.1, is applied similarly to each training or test digit image for normalization purposes. The normalized digit image is then submitted to the feature extractors to perform convolution using the 24 generated Gabor filters, where the *App-1* or *App-2* then takes place for feature selection to compose the final feature vector of the digit sample.

The first approach *App-1* is concerned with applying the first technique of dimensionality reduction and feature selection using *Algorithm-1* to extract 704-value Gabor-based feature vector, as described in Section 4.1.1 and recapitulated in Table 4. Thereafter, these 704 features are concatenated with the four supplementary features (*Ftr1*, *Ftr2*, *Ftr3*, and *Ftr4*) presented in Section 4.2. This ends with a total of 708 values representing a single *App-1* feature vector. On the other hand, the second approach *App-2* is associated with applying the second technique of dimensionality reduction along with feature selection using *Algorithm-2* to extract a 78-value Gabor-based feature vector, as described in Section 4.1.2 and outlined in Table 4. Here the extracted 78 features are also concatenated with the same four supplementary features (i.e. *Ftr1*, *Ftr2*, *Ftr3*, and *Ftr4*). As such, a total of 82 values are used to represent a single *App-2* feature vector. Eventually, the resulting 708-value feature vector of *App-1* and the 82-value feature vector of *App-2* are used to evaluate the recognition performance with respect to each of the three classifiers on both databases.

In overview, as shown in Table 4, the performance of all approaches with all three classifiers when tested on DB1 is better than the performance of their counterparts tested on DB2. Besides, the *k*-NN classifiers yield higher performance than FFNN classifier, particularly when tested on DB2. The overall results of *App-1* apparently offer better performance by all accounts than *App-2*, though that *App-2* still offers very similar high performance (especially on DB1) with only 82 features, about one ninth of *App-1* features, that significantly reduce the computation time of the digit recognition task.

The reported performance results in Table 4 also show that 3-NN classifier attains the highest performance when used with either *App-1* by a score of 99.87% or *App-2* by a score of 99.69%. However, the highest recognition rates per database are 99.87% and 98.30% achieved by *App-1* when used with 3-NN classifier on both DB1 and DB2 respectively. Furthermore, this approach also obtains the highest average accuracy of 99.09% amongst all other methods.

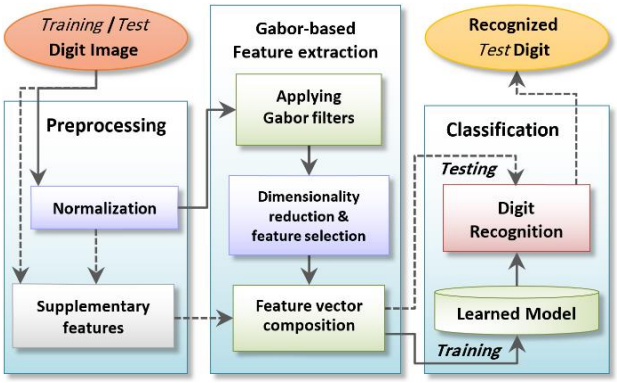



Fig. 5. Overview of the Digit Recognition System.

TABLE IV. CONDUCTED EXPERIMENTS AND THEIR RESULTS FOR EACH GABOR-BASED APPROACH

| Approach | Gabor-based feature extraction and selection | | Supplementary features | Total features | Classifier | Accuracy | | |
|---|---|---|---|---|---|---|---|---|
| | Dimensionality reduction | Feature vector composition | | | | DB1 | DB2 | Avg. |
| ***App-1*** | The 4x6 Gabor convolution, comprising 128×192 feature matrix is reduced to a smaller 22×32 matrix using ***Algorithm-1*** | The 22×32 reduced matrix is used as an orthogonal 704-value Gabor-based feature vector | ***Ftr1***, ***Ftr2***, ***Ftr3***, and ***Ftr4*** | 708 | 1-NN | **99.87** | 98.26 | 99.07 |
| | | | | | 3-NN | **99.87** | **98.30** | **99.09** |
| | | | | | FFNN | 99.84 | 91.27 | 95.56 |
| ***App-2*** | The 4x6 Gabor convolution, comprising 128×192 feature matrix is reduced to a smaller 26×39 matrix using ***Algorithm-2*** | The mean and the mean of the covariance matrix are computed for the 26×39 reduced matrix, resulting in a 39-value vector each, then concatenated as a 78-value Gabor-based feature vector | ***Ftr1***, ***Ftr2***, ***Ftr3***, and ***Ftr4*** | 82 | 1-NN | 99.67 | 98.19 | 98.93 |
| | | | | | 3-NN | 99.69 | 98.10 | 98.90 |
| | | | | | FFNN | 99.16 | 76.62 | 87.89 |

Table 5 and Table 6 show the confusion matrices inferred for the best recognition performance on DB1 and DB2 respectively, which achieved using *App-1* with 3-NN.

It is noteworthy that *App-1* attains the same highest accuracy of 99.87% with both 1-NN and 3-NN classifiers when applied on DB1. Hence, only 8 out of 6160 test digit samples are misclassified by each classifier of 1-NN and 3-NN when used with *App-1*. Fig. 6 shows the misclassified test digit samples from DB1 using *App-1* with each of 1-NN and 3-NN respectively. With regard to DB2, 170 out of 10000 digit samples in the test dataset are misclassified using *App-1* with 3-NN, which presents lower accuracy and more errors than what achieved on DB1, since that DB2 is considerably a larger database and consequently consists of a larger number of likely unrecognizable digit samples, as expected.

TABLE V. CONFUSION MATRIX OF THE RECOGNITION PERFORMANCE ON **DB1** USING *APP-1* WITH 3-NN

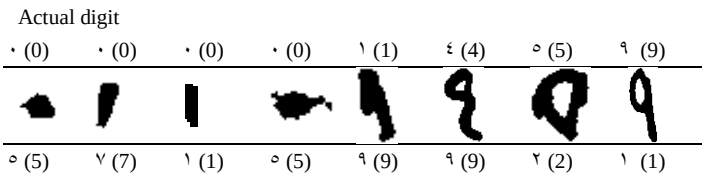
| | **Predicted digit** | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Actual digit** | **#** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| | **0** | **611** | 1 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 1 |
| | **1** | 0 | **616** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **2** | 0 | 0 | **616** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **3** | 0 | 0 | 0 | **616** | 0 | 0 | 0 | 0 | 0 | 0 |
| | **4** | 0 | 0 | 0 | 0 | **615** | 1 | 0 | 0 | 0 | 0 |
| | **5** | 0 | 0 | 1 | 0 | 0 | **615** | 0 | 0 | 0 | 0 |
| | **6** | 0 | 0 | 0 | 0 | 0 | 0 | **616** | 0 | 0 | 0 |
| | **7** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **616** | 0 | 0 |
| | **8** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **616** | 0 |
| | **9** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **615** |

TABLE VI. CONFUSION MATRIX OF THE RECOGNITION PERFORMANCE ON **DB2** USING *APP-1* WITH 3-NN

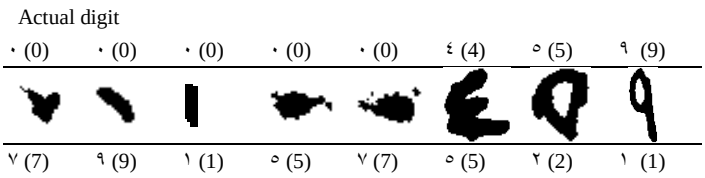
| | **Predicted digit** | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Actual digit** | **#** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| | **0** | **919** | 8 | 4 | 1 | 1 | 59 | 0 | 1 | 3 | 4 |
| | **1** | 2 | **995** | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | **2** | 1 | 0 | **997** | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| | **3** | 0 | 2 | 14 | **982** | 0 | 0 | 1 | 1 | 0 | 0 |
| | **4** | 0 | 4 | 6 | 0 | **990** | 0 | 0 | 0 | 0 | 0 |
| | **5** | 3 | 0 | 8 | 0 | 1 | **981** | 0 | 1 | 6 | 0 |
| | **6** | 0 | 2 | 0 | 0 | 0 | 0 | **997** | 0 | 0 | 1 |
| | **7** | 1 | 1 | 0 | 0 | 0 | 3 | 1 | **994** | 0 | 0 |
| | **8** | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | **996** | 1 |
| | **9** | 0 | 12 | 0 | 0 | 4 | 0 | 4 | 0 | 1 | **979** |

A further analysis to the recognition performance and consequent misclassifications caused using the various adopted techniques reveal that DB1 and DB2 contain a number bad and distorted data samples or deformed digit strokes (might be confusing or even unrecognizable by human), which no doubt increases the challenge and produces more classification errors. Fig. 7 shows some examples of such challenging and confusing digit samples.

Digit '0' consistently receives the highest error rates in all conducted experiments. Therefore, it can be observed in Fig. 6 that about 50% to 60% of the misclassified DB1 samples actually belong to digit (or class) '0'. Furthermore, around 48% of DB2 misclassifications are also samples for digit '0'. It can be noticed that digit '0' is the only digit in DB2 that is confused with all other digits but digit '6', while it is mostly confused with digit '5' (causing 59 misclassification instances) followed by digit '1' (causing more eight instances), as shown in Table 6.

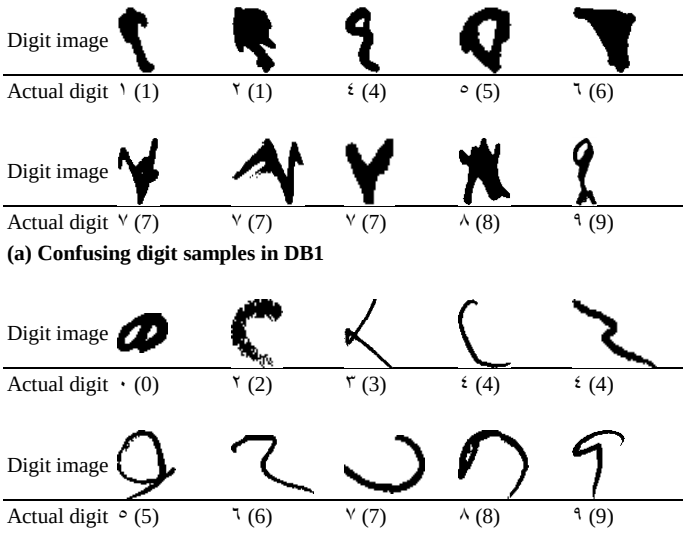



**(a) Only 8 misclassified DB1 samples (using *App-1* & 1-NN)**




**(b) Only 8 misclassified DB1 samples (using *App-1* & 3-NN)**

Fig. 6. Misclassified Test Samples from **DB1**.




**(a) Confusing digit samples in DB1**

**(b) Confusing digit samples in DB2**

Fig. 7. Examples of Challenging Digit Samples.

This could be justified by the fact that the nature of digit '0' shape in case of Arabic-Indic format, in which it is represented by computer typing/printing as '0' and is typically written by hand as a little dot resulting in a verity of random shapes of that little dot. Such random shapes of digit '0' are more likely in case of fast handwriting, which increases the probability of producing more unintended other digit-alike samples enabling confusion with almost all other digits [18]. This in turn is considered as a major challenge to be confronted by size invariant digit recognition systems.

## VII. Conclusions and Discussions

In this work, we propose two Gabor-based approaches *App-1* and *App-2* using different feature subset selection algorithms for Arabic-Indic handwritten digit recognition system. Each approach is examined in turn with three adopted classifiers 1-NN, 3-NN, and FFNN for recognition performance evaluation. For the sake of performance variability assessment, validation, and comparison, these experiments are applied to two different benchmark databases DB1 and DB2.

The obtained performance results show high recognition rates for both proposed approaches ranging from 99.69% to 99.87% as the best achieved accuracies. Regardless of used databases, the *k*-NN classifiers attain higher performance than FFNN classifier in general; and more apparent when tested on DB2.

The overall results of *App-1* obviously offer better performance by all accounts than *App-2*, though that *App-2* still offers very similar high performance (especially on DB1) with only 82 features –about one ninth of *App-1* features– that significantly shorten the complexity time and make *App-2* more suited and functional approach for online handwritten digit recognition systems.

For future, the proposed system handwritten Arabic-Indic digit recognition can be further extended and applied for handwritten Arabic character and text recognition. Furthermore, the same methodology is likely to be enforced with minor modifications in potential typewritten or handwritten OCR systems for other languages.

## References

[1] A. Radaideh and M. S. M. Rahim, "Existing techniques in Arabic characters recognition (ACR)," Journal of Informatics and Mathematical Sciences, vol. 8, no. 5, 2016, pp. 347-360.

[2] P. Yadav and N. Yadav, "Handwriting recognition system-a review," International Journal of Computer Applications, vol. 114, no. 19, March 2015, pp. 36-40.

[3] H. Modi and M. Parikh, "A review on optical character recognition techniques," International Journal of Computer Applications, vol. 160, no. 6, 2017, pp. 20-24.

[4] K. S. Dash, N. B. Puhan, and G. Panda, "Handwritten numeral recognition using non-redundant Stockwell transform and bio-inspired optimal zoning," IET Image Processing, vol. 9, no. 10, 2015, pp. 874-882.

[5] M. O. Assayony and S. A. Mahmoud, "An enhanced bag-of-features framework for Arabic handwritten sub-words and digits recognition," Journal of Pattern Recognition and Intelligent Systems, vol. 4, no. 1, 2016, pp. 27-38.

[6] M. A. Radwan, M. I. Khalil, and H. M. Abbas, "Neural networks pipeline for offline machine printed Arabic OCR," Neural Processing Letters, vol. 48, no. 2, 2018, pp. 769-787.

[7] A. Zaafouri, M. Sayadi, and F. Fnaiech, "A vision approach for expiry date recognition using stretched Gabor features," The International Arab Journal of Information Technology, vol. 12, no. 5, 2015, pp. 448-455.

[8] N. Karayiannis and S. Behnke, "New radial basis neural networks and their application in a large-scale handwritten digit recognition problem," in Recent advances in artificial neural networks: CRC Press, 2018, pp.61-116.

[9] C. Kaensar, "Analysis on the parameter of back propagation algorithm with three weight adjustment structure for hand written digit recognition," in 10th International Conference on Service Systems and Service Management (ICSSSM), 2013, pp. 18-22: IEEE.

[10] C. Kaensar, "A comparative study on handwriting digit recognition classifier using neural network, support vector machine and k-nearest neighbor," in 9th International Conference on Computing and Information Technology (IC2IT), 2013, pp. 155-163: Springer.

[11] E. Tuba, M. Tuba, and D. Simian, "Handwritten digit recognition by support vector machine optimized by bat algorithm," in 24th Conference on Computer Graphics, Visualization and Computer Vision (WSCG), 2016, pp. 369-375.

[12] T. N. Do and N. K. Pham, "Handwritten digit recognition using GIST descriptors and random oblique decision trees," in Some Current Advanced Researches on Information and Computer Science in Vietnam: Springer, 2015.

[13] S. Arya, I. Chhabra, and G. S. Lehal, "Recognition of Devnagari numerals using Gabor filter," Indian Journal of Science and Technology, vol. 8, no. 27, 2015, pp. 1-6.

[14] Q. Abbas, W. H. Bangyal, and J. Ahmad, "Analysis of learning rate using BP algorithm for hand written digit recognition application," in International Conference on Information and Emerging Technologies (ICIET), 2010: IEEE.

[15] H. Alamri, C. L. He, and C. Y. Suen, "A new approach for segmentation and recognition of Arabic handwritten touching numeral pairs," in International Conference on Computer Analysis of Images and Patterns, 2009, pp. 165-172: Springer.

[16] M. Kherallah, A. Elbaati, H. E. Abed, and A. M. Alimi, "The on/off (LMCA) dual Arabic handwriting database," in Proceedings of the 11th International Conference on Frontiers in Handwriting Recognition, 2008.

[17] I. A. Doush, F. Alkhateeb, and A. H. Gharaibeh, "A novel Arabic OCR post-processing using rule-based and word context techniques," International Journal on Document Analysis and Recognition (IJDAR), vol. 21, no. 1-2, 2018, pp. 77-89.

[18] S. Abdleazeem and E. El-Sherif, "Arabic handwritten digit recognition," International Journal of Document Analysis and Recognition (IJDAR), vol. 11, no. 3, 2008, pp. 127-141.

[19] A. Ashiquzzaman and A. K. Tushar, "Handwritten Arabic numeral recognition using deep learning neural networks," IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR), 2017.

[20] A. Lawgali, "Handwritten digit recognition based on DWT and DCT," International Journal of Database Theory and Application, vol. 8, no. 5, 2015, pp. 215-222.

[21] E. A. El-Sherif and S. Abdelazeem, "A two-stage system for Arabic handwritten digit recognition tested on a new large database," in Artificial Intelligence and Pattern Recognition, 2007, pp. 237-242.

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

[23] M. Ramzan et al., "A survey on using neural network based algorithms for hand written digit recognition," International Journal of Advanced Computer Science and Applications, vol. 9, no. 9, 2018, pp. 519-528.

[24] S. Naz, S. B. Ahmed, R. Ahmad, and M. I. Razzak, "Arabic script based digit recognition systems," in International Conference on Recent Advances in Computer Systems (RACS), 2016, pp. 67-73.

[25] S. Luan, C. Chen, B. Zhang, J. Han, and J. Liu, "Gabor convolutional networks," IEEE Transactions on Image Processing, vol. 27, no. 9, 2018, pp. 4357-4366.

[26] E. S. Jaha and L. Ghouti, "Color face recognition using quaternion PCA," in 4th International Conference on Imaging for Crime Detection and Prevention (ICDP), IET, 2011.

[27] S. S. Sarwar, P. Panda, and K. Roy, "Gabor filter assisted energy efficient fast learning convolutional neural networks," in IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), 2017.

[28] J.-K. Kamarainen, V. Kyrki, and H. Kalviainen, "Invariance properties of Gabor filter-based features-overview and applications," IEEE Transactions on image processing, vol. 15, no. 5, 2006, pp. 1088-1099.

[29] E. S. Jaha and M. S. Nixon, "From clothing to identity: manual and automatic soft biometrics," IEEE Transactions on Information Forensics and Security, vol. 11, no. 10, 2016, pp. 2377-2390.

[30] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern classification. John Wiley & Sons, 2012.

[31] E. Fiesler and R. Beale, "Neural network topologies," The Handbook of Neural Computation, E. Fiesler and R. Beale (Editors-in-Chief), Oxford University Press and IOP Publishing, 1996.

[32] A. Heiat, "Comparison of artificial neural network and regression models for estimating software development effort," Information and software Technology, vol. 44, no. 15, 2002, pp. 911-922.