Signal and Image Processing Letters
Vol. 1., No. 2, July 2019, pp. 41-49
ISSN 2714-6677
https://simple.ascee.org/

Comparative Study of KNN, SVM and SR Classifiers in Recognizing Arabic Handwritten Characters based on Feature Fusion

Abu Sayeed A. Huque[a,1,*], Mainul Haque [b,2], Haidar A. Khan [b,3], Abdullah Al Helal [b,4], Khawza I. Ahmed [b,5]

[a] University of Tabouk, Tabouk, Saudi Arabia
[b] United International University, Dhaka, Bangladesh
[1] abusayeed.h@gmail.com*; [2] mainul37@gmail.com; [3] hkhan@eee.uiu.ac.bd; [4] helal@eee.uiu.ac.bd; [5] khawza@eee.uiu.ac.bd
* corresponding author

ARTICLE INFO

Article history
Received February 18, 2019
Revised March 18, 2019
Accepted July 19, 2019

Keywords
Arabic OCR
K-Nearest Neighbor
Support Vector machine
Sparse Representation Classifier
Fusion

ABSTRACT

This paper evaluates and compares the performance of K-Nearest Neighbors (KNN), Support Vector Machine (SVM) and Sparse Representation Classifier (SRC) for recognition of isolated Arabic handwritten characters. The proposed framework converts the gray-scale character image to a binary image through Otsu thresholding, and size-normalizes the binary image for feature extraction. Next, we exploit image down-sampling and the histogram of image gradients as features for image classification and apply fusion (combination) of these features to improve the recognition accuracy. The performance of the proposed system is evaluated on Isolated Farsi/Arabic Handwritten Character Database (IFHCDB) – a large dataset containing gray scale character images. Experimental results reveal that the histogram of gradient consistently outperforms down-sampling based features, and the fusion of these two feature sets achieves the best performance. Likewise, SRC and SVM both outperform KNN, with the latter performing the best among the three. Finally, we achieved a commanding accuracy of 93.71% in character recognition with fusion of features classified by SVM, where 92.06% and 91.10% is achieved by SRC and KNN respectively.

1. Introduction

Optical character recognition (OCR) is an important application of machine learning. It is a process of identifying the text in a digital image and saving the text characters in an electronic file. OCR has many usages, such as, mail sorting, automatic number plate recognition, extracting business card information into a contact list, data entry, etc. Among them, a very useful application of OCR is handwritten character recognition.

In order to achieve higher recognition rate and reduce the processing time, many new methods and techniques have been developed by the researchers. So far, most of the works have been done on English, Latin, and Chinese character classification; while, Arabic character recognition has been somewhat neglected. However, in recent years, the recognition of Arabic handwritten character is drawing increasing attention [1][2]. More than 400 million people speak Arabic language [3]. Arabs as well as 23 countries in the Middle East and North African countries speak Arabic. This language has a very rich vocabulary and the script is written from right to left. There may be several shapes for a single Arabic character depending on its position in a word. In other words, the shape of a character varies if it is at the beginning or in the middle or at the end of a word, as well as, in its appearance in isolation in a sentence as depicted in Fig. 1. This variation exists even in the numerals. For instance, the digit 4, 5 and 6 have two different shapes that are shown in Fig. 2.

At end	In middle	At beginning	Isolated
ـف	ـفـ	فـ	ف

Fig. 1. Different shapes in different position for an Arabic character.

0	1	2	3	4	5	6	7	8	9
٠	١	٢	٣	٤	٥	٦	٧	٨	٩
				۴	۵	۶			

Fig. 2. English and Arabic Digits.

Lawgali, Buridane *et al*. [4] created their own Arabic handwritten database containing 5600 characters. They applied Discrete Cosine Transform (DCT) based feature set on it and achieved 96.56% accuracy.

Addakiri and Bahaji [5] used Neural Network technique and obtained the average accuracy of 83%. The system was tested against 1400 different characters written by ten users. An online based system using Neural Network was introduced by [6] for isolated Arabic handwritten character. In their experiment, they collected data from different users by using an external mouse. Each user was asked to write 104 isolated different Arabic characters. The system obtained 95.7% and 99.1% accuracy for untrained and trained writers respectively. However, Zawaideh [7] had implemented a modified Multi-Neural Network classifier. The proposed system is evaluated on a database of 100 different isolated characters that are written by 10 different writers in "Roq'a" style Arabic character. Based on the character shape, the system results vary between 51%-77%. Parvez and Mahmoud [8] had also presented a novel method for recognizing isolated alphanumeric handwritten Arabic characters. The experiment was performed using a Nearest Neighbor (NN) classifier which is based on fuzzy attributed tuning function (FATF). In this paper, two databases were used; one is for the characters and the other is for the numerals. The character database contains 1948 samples written by 4 writers. On the other hand, the numerals database, known as ADBase, contains 70,000 digits written by 700 writers. Accuracy rate of 98% and 97% have been obtained by the proposed system for characters and numerals respectively.

In addition, Saidni, Asma *et al*. [9] exploited Histogram of Oriented Gradients (HOG) descriptor for machine-printed and handwritten Arabic word recognition, and achieved 99.07% accuracy for word classification. Furthermore, Biglari, Mohsen *et al*. [10] used Local Binary Pattern (LBP) operator as features with a multi-layer perceptron neural network for Arabic handwritten digit recognition and achieved 99.72% accuracy for digit classification.

This paper focuses on the recognition of isolated Arabic character, both numerals as well as alphabets. To evaluate this, a rich Arabic database is a must, but finding a dependable Arabic database has become one of the biggest challenges. There are some handwriting recognition systems that manifest high accuracy rate due to the fact that they were tested against small databases [11]. Many of the databases were created from segmented words [12]. In order to prove the robustness of the system, a large standard database is required. Unfortunately, only a limited number of databases were developed for Arabic handwritten characters.

We have tested our recognition method on Isolated Farsi/Arabic Handwritten Character Database (IFHCDB) [13]. This database contains all isolated characters. This database is mostly used for digit recognition though it has both alphabets and numerals [14]. In our experiment, we have used the database in its entirety. We have exploited K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Sparse Representation Classifier (SRC). KNN is a simple instance-based classifier, and SVM is a state of the art classifier. Therefore, we wanted to compare the performance of SRC, a relatively novel classifier, against other more traditional classifiers. To the best of the authors' knowledge, SRC is applied for the first time to Arabic handwritten character recognition. We have also applied feature fusion method in an attempt to get improved outcome [15][16].

The paper is organized in six sections as follows. Section II describes the methods of pre-processing and feature extraction. Section III briefly discusses about the K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Sparse Representation Classifier (SRC). Comparative analysis of experiments and results are captured in Section IV and V, respectively. Finally, Section VI concludes the paper with final remarks.

2. Methods

There are several stages in the Optical Character Recognition (OCR) system. The stages are pre-processing, feature extraction, and classification. Fig. 3 demonstrates the schematic diagram of the character recognition system.

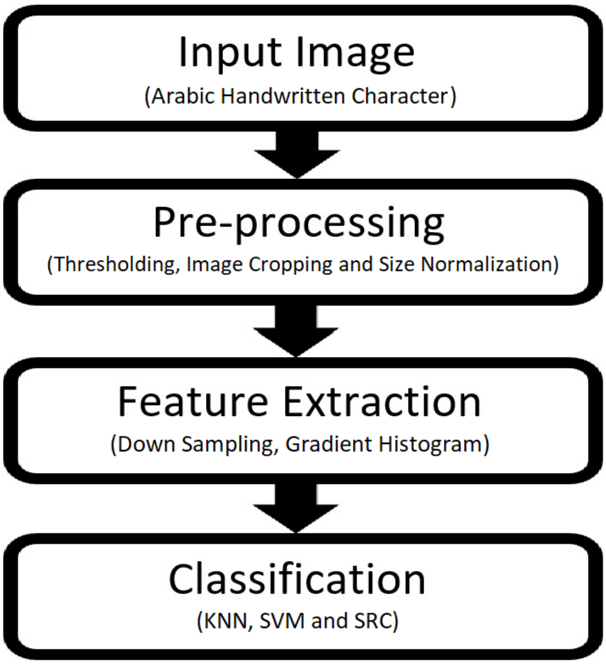


Fig. 3. Block diagram of character recognition system.

2.1. Pre-Processing

The pre-processing plays a very important role in achieving improved result in any (pattern) recognition system. A series of operations, touched upon briefly in the following subsections, are performed on scanned input image in pre-processing task. The aim of pre-processing is to enhance the readability of the character image and remove those details that have no contribution in the process of recognition.

A. Thresholding

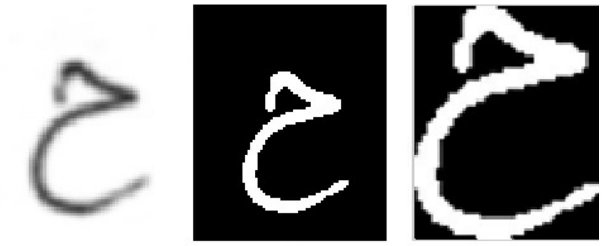
The first step in image pre-processing is the thresholding. In thresholding, the image, color or gray, is converted into binary. There are two types of thresholding, namely, global [17] and local. In global image thresholding, there is only one threshold value; whereas in local image thresholding, different threshold values for different image segments. We apply global image thresholding on the images. Otsu's thresholding method [18] is used to select a global threshold. Otsu's method selects a global threshold such that the intra-class (within class) variance is minimized, and as a result, the inter-class (between class) variance is maximized [18]. It separates the character from its background as well as helps reduce noise. An example of such improvement is shown in Fig. 4(b). During the image scanning process, usually some distortion is introduced to the image. Small part that is not part of the writing can be considered as noise and this noise is also removed by the same thresholding operation.

B. Image Cropping

Image cropping is the next step for pre-processing. Cropping means removing some irrelevant parts of the image so that the image's region of interest can be focused. Depending on the application, the method usually consists of the removal of some of the peripheral areas of an image to get rid of extraneous trash from the image, to enhance its framing, to vary the ratio, or to intensify or isolate the subject matter from its background.

C. Size Normalization

In normalization part, all images are normalized to a standard size. It is done by either down sampling or up-sampling of the input image. We use down-sampling and the character images are normalized to 32×32 pixels size as presented in Fig. 4(c).



a b c

Fig. 4. Illustration of pre-processing (a) original image, (b) Otsu thresholding image, and (c) normalized image.

2.2. Feature Extraction

The feature extraction stage is used to extract the most relevant information from the pre-processed character image which helps recognize the exact character. As it is the most important stage in the process, the optimum performance of this stage ensures improved recognition rate, and hence, reduces the misclassification. The recognition of exact character heavily depends on the selection of feature set. There are different types of features [19][20], such as, structural features, statistical features, moment-based features, and global transformation-based features etc. However, in this paper, we used down sampling and gradient based directional features.

A. Down Sampling

In down sampling, size of the original images is reduced to lower scale. In our experiment, pre-processed images are down sampled to 4×4 and 8×8 pixels image size. Here each pixel acts as a feature. Therefore, the down sampled images have 16 and 64 features respectively. Moreover, down sampling also acts as a low pass filter or mean filter that helps reduce the effect of minor differences in the character image that are caused by inter-user variability. It is relevant to mention here that the image is first pass through a bi-cubic interpolation block, acting as a low pass filter, to band limit the signal appropriately so that no aliases appear after the down sampling process.

B. Gradient Histogram

The most common features used by the researchers are extracted from the evaluation of the local stroke orientation/direction distribution by the gradient histogram [21]. To evaluate the gradient features, at first the input images are to be normalized into 32×32 pixels which we determined in the pre-processing stage. The gradient vector is calculated from normalized input image by using Sobel operator. Then the vector at each pixel is assigned to discrete directions by parallelogram decomposition. We use 8-directional gradient histogram as it produces better result than the result obtained by using 4, 12, or 16 directional gradient histograms. Now this vector is down sampled to 4×4 and 8×8 for 8 directions separately and, as a result, 128 ($4\times4\times8$) and 512 ($8\times8\times8$) gradient features are obtained respectively.

3. Classification

The classification is the main decision making stage of a recognition system. In this stage, the features, which have been extracted in the previous stage, are exploited to identify the characters according to the pre-set rules. In this paper, we have evaluated the performance of three classifiers, which are K-Nearest Neighbor (KNN), Support Vector Machines (SVM), and Sparse Representation Classifier (SRC) by using different feature set, namely, down sampling, gradient histogram and the concatenation of these two.

3.1. K-Nearest Neighbor (KNN)

K-nearest neighbor is one of the most well-known and widely used classifiers. The algorithm of this classifier is amongst the simplest of all machine learning algorithms [22]. In KNN for a test sample, usually classification decision is made by choosing majority voting [23] from the obtained *K* nearest neighbors data. The performance of KNN mostly depends on the proper choice of a parameter called *K* and the distance metric, where *K* is a positive integer. In this paper, we used Euclidean distance for finding the nearest neighbor and varied *K* from 1 to 10 for classification.

3.2. Support Vector Machine (SVM)

Support vector machine is another popular classifier. It is a group of supervised learning methods [24]. In SVM classification, dataset is separated into training and testing sets as a default practice used in all pattern recognition methods. This classifier exploits the training set to generate a model which predicts the target values of the test data when only the test data attributes are given. It is a binary classifier [25]. To split the training set into two classes a hyperplane is placed. The hyperplane is placed in a position that has the maximum distance between data points of both classes. This kind of classification is known as linear classification. However, in non-linear classification training data are not always linearly dividable. To perform the non-linear classification, kernel functions, an important factor in SVM, are used. These kernel functions map the original space into an upper dimensional space for smooth separation [26]. Though there exist different types of kernel functions, such as, Linear kernel, Polynomial kernel, Radial Basis Function (RBF) and Sigmoid, in our experiment, we have preferred to use Radial Basis Function (RBF).

3.3. Sparse Representation Classifier (SRC)

Sparse Representation Classifier is based on the non-conventional compressive sensing principle. This is a recently developed classification technique. To recognize a character image $I(x,y)$ from n_c number of classes in SRC, each image $I_i(x,y)$ is represented by its feature vector $b \in \Re^m$ in a vector space, where m is number of features. For n number of classes $A \in \Re^{mn}$ formed. Where A can be computed using (1).

$$A = [b_1, b_2, \dots, b_n] \tag{1}$$

In our experiment, it is assumed any character image can be approximately represented as a linear combination of the images of its native class because of considering errors and noises in data. The equation for any test sample b as a linear combination of the training sample is shown in (2) - (4).

$$b \approx a_1 b_1 + a_2 b_2 + \cdots + a_n b_n = A x_r \tag{2}$$

or

$$A x_r = b + e \tag{3}$$

where $e = A x_r - b$ is noise term with $\|e\|_2 < \epsilon$

$$x_r = [a_1, a_2, \dots, a_n]^T \Re^n \epsilon \tag{4}$$

Thus, we have to find x_r, the vector of coefficients a_i. The linear system (2) cannot be solved by traditional approach. If m=n, (2) has a unique solution, implying a unique classification. In general m is not equal to n. Therefore, if $m < n$, (2) is an under-determined system, implying infinite solution. In the case of $m > n$, (2) is an over-determined system, which means there is no solution. However, for over-determined system, considering the modelling and measurement error, there exist methods to find an approximate solution to (2).

Thus, the representation of any test sample in (2) will have many coefficients as zero, which leads to sparse representation [27] [28]. Sparse representation of the test sample motivates us to choose, out of infinite solutions, the one which is the sparsest, *i.e.*, with most zeros (see (5)).

$$x_0 = \arg\min_x \|x\|_0, \quad \text{subject to } \|Ax - b\|_2 < \epsilon, \tag{5}$$

Where $\|x\|_0$ indicates the number of non-zero elements in x. Though (5) requires exhaustive search, for *sparse enough* x_r, (5) is equivalent to (6).

$$x_1 = \arg\min_x \|x\|_1, \quad \text{subject to } \|Ax - b\|_2 < \epsilon, \tag{6}$$

Where $\|x\|_1$, indicates the absolute sum of elements in x. Therefore, the solution [29] is shown in (7).

$$x_{l_1} = \arg\min_x (\|Ax - b\|_2 + \lambda \|x\|_1). \tag{7}$$

4. Experiment

We have evaluated the proposed systems on Isolated Farsi/Arabic Handwritten Character Database (IFHCDB) [13] created at the Electrical Engineering Department of Amirkabir University of Technology (AUT), Tehran, Iran in 2006. It is one of the standard database which contains a large number of Arabic (and Farsi) handwritten character images. Most of the other Arabic database is either small or created for specific purpose, such as, cheque authentication [30], postal address determination [31] etc. The IFHCDB database contains gray scale with a resolution of 300 dpi images of 52,380 characters and 17,740 numerals in total. The dimensions of images are *77×95* pixels. There are 47 classes. Fig. 5 and Fig. 6 show some character and numeral samples in the database respectively. The number of samples in each class is non-uniform. The database is divided into a test set and a training set by the developer as a standard procedure practiced in pattern recognition systems. The test set contains 30% images, while the training set contains 70% images. At first we pre-processed the images of the database. From this pre-processed images, we extract down sampled images and gradient histogram feature set. Then feature fusion is done thereafter. For classification, we used the training set to develop algorithm for KNN, SVM, and SRC; while used the test set for testing. In KNN, we select distance metric as Euclidean distance for classification. As hinted in section 3 (Classification), we varied the value of K from 1 to 10. However, we choose K as 1 (for simplicity purposes) and 9 (for producing the best result). We know that SVM is a binary classifier, whereas, our dataset contains more than two classes. To tackle this issue, we used a library for SVM known as LIBSVM [32] that supports multi-class [33] classification. For recognition, Radial Basis kernel Function is used here. For SRC classifier, in (7), we varied λ, known as a regularization parameter, which controls the effect of the dynamic model on the solution. We swept λ's value from 0.1 to 1 in 0.1 increment to obtain the best result from SRC classifier.

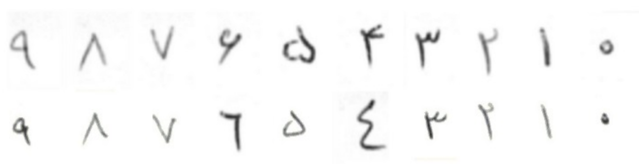


Fig. 5. Arabic Handwritten numeral samples.

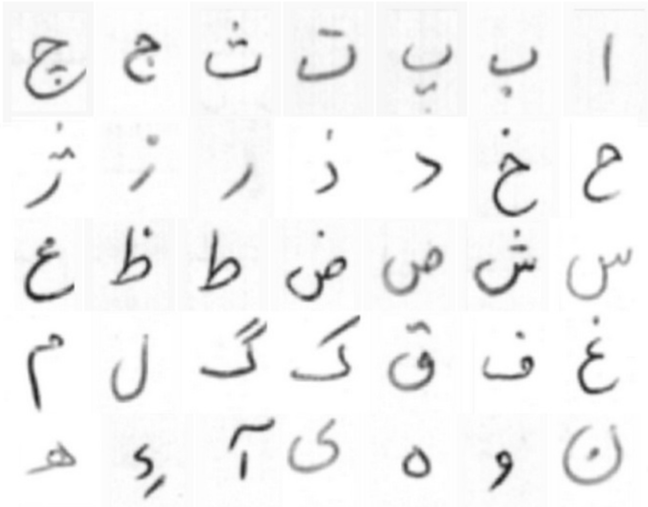


Fig. 6. Arabic Handwritten character samples.

5. Results

The accuracy percentage of two different feature sets and three classifiers are shown in Table 1. It is observed that SVM provided the best result amongst all other classifiers for different feature sets. It is also observed that the fusion (concatenation) of down sampling and gradient histogram feature sets demonstrate the best recognition rate for all classifiers.

As shown in Table 1, for the 4×4 image size, the gradient histogram and concatenation (fusion) of features provide the higher accuracy for all classifiers. The accuracy is more than 90.75%. Contrastingly, the down sampling based feature produce better accuracy (over 90.05%) for 8×8 image size. The outcome exhibits that down sampling based feature works better for bigger image size while other features seem more suitable for smaller images. However, for all image size, the fusion of down sampling & gradient histogram feature achieves the highest accuracies for all classifiers. The best character recognition accuracy is 93.71% produced by the SVM classifier for 4×4 image size. These results indicate that fusion of features works better in comparison with the scenarios when the features are used independently.

Table 1. Overall accuracy for character recognition

Classifier	Image size	Down Sampling	Gradient Histogram	Fusion Features
KNN (K=1)	4x4	85.26	88.94	89.27
	8x8	87.80	88.13	88.28
KNN (K=9)	4x4	88.20	90.75	91.10
	8x8	90.05	89.78	90.18
SVM	4x4	89.19	93.68	93.71
	8x8	92.12	93.22	93.25
SRC	4x4	79.14	92.04	92.06
	8x8	90.89	91.52	91.65

6. Conclusion

The main objective of this experiment is to recognize the isolated Arabic character. We have applied variety of features and classifiers to achieve it. Our experiments revealed that both SRC and SVM consistently outperformed KNN, while SVM achieved the highest recognition rate. Similarly, histogram of image gradient achieved better recognition rate than that of down-sampling based features, and the fusion of these two feature sets achieved the best performance. The fact that SVM outperforms all other classifiers for different feature sets should not be strikingly surprising. KNN is a simple classifier. Euclidean distance is also not the best metric for distance between two images (e.g. two identical but slightly sifted images will have a huge Euclidean distance). Thus, it is intuitive that both SRC and SVM would outperform KNN. SRC reconstructs the test (or unknown) image as a linear combination of training images. While this approach achieves improved performance, SVM has the advantage of transforming features into a higher dimension for better classification. Analyzing features in a higher dimensional space helps SVM achieve even better performance.

It is to be noted here that many other researchers also used this same dataset, which consists of both numerals (12) as well as alphabets (35). However, to the best of our knowledge, none of them used the entire dataset like we did. Early researchers focused only on the digits, therefore their comparison set was limited to 12 members in the dataset, while the later researchers turned their attention to the alphabets alone, and again, used part of the dataset for comparison purpose. For instance, the letter alif and the digit 1 in handwritten Arabic are almost identical. So, the first group cannot recognize a 1 as alif mistakenly because alif is not even in their consideration. Likewise, the second group does not mistakenly recognize an alif as 1 since 1 is not even in their consideration. Naturally, their accuracy would be better than who takes in the entire dataset for comparison purpose to identify numerals as well as alphabets. Therefore, it is unfair to compare their partial accuracy results with ours.

In future, we would like to perform additional image preprocessing such as skew correction and slant removal, and exploit feature sets such LBP and Chain Codes to evaluate performance of classifiers, such as, Neural Network and Random Forest etc.

Acknowledgment

The authors would like to thank the Deanship of Scientific Research in University of Tabuk for their financial support (grant no. S0282-1436).

References

[1] A. A. A. Ali and M. Suresha, "Arabic handwritten character recognition using machine learning approaches," in *2019 Fifth International Conference on Image Information Processing (ICIIP)*. IEEE, 2019, pp. 187–192.
[2] S.N. Srihari and G. Ball, Guide to OCR for Arabic Scripts, Springer London, 2012.
[3] B. E. Sawe, "What is the Most Spoken Language in the World?," *WorldAtlas*, Jun. 7, 2019. [Online]. Available: https://www.worldatlas.com/articles/most-popular-languages-in-the-world.html. [Accessed: Mar. 08, 2020].
[4] A. Lawgali, A. Bouridane, M. Angelova and Z. Ghassemlooy, "Handwritten Arabic character recognition: Which feature extraction method?," *International Journal of Advanced Science and Technology*, vol. 34, Sep. 2011, pp. 1-8.
[5] K. Addakiri and M. Bahaj. "On-line Handwritten Arabic Character Recognition using Artificial Neural Network," *International Journal of Computer Applications*, vol. 55, no. 13, Oct. 2012, pp. 42-46.
[6] B. Alijla and K. Kwaik, "Oiahcr: online isolated arabic handwritten character recognition using neural network," *Int. Arab J. Inf. Technol.*, vol. 9, no. 4, July 2012, pp. 343-351.
[7] F.H. Zawaideh, "Arabic handwritten character recognition using modified multi-neural network," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 3, no. 7, July 2012, pp. 1021-1026.
[8] M.T. Parvez and S. Mahmoud, "Arabic Handwritten Alphanumeric Character Recognition using Fuzzy Attributed Turning Functions", *Proc. Workshop in Frontiers in Arabic Handwriting Recognition, in conjunction with 20th ICPR*, Istanbul, Turkey, Aug. 2010, pp. 9-14.
[9] A. Saidani, A. Kacem Echi, and A. Belaid, "Arabic/latin and machineprinted/handwritten word discrimination using hog-based shape descriptor," *ELCVIA: electronic letters on computer vision and image analysis*, pp. 0001–23, 2015.
[10] M. Biglari, F. Mirzaei, and J. G. Neycharan, "Persian/arabic handwritten digit recognition using local binary pattern," *International Journal of Digital Information and Wireless Communications (IJDIWC)*, vol. 4, no. 4, pp. 486–492, 2014.
[11] K. S. Younis, "Arabic handwritten character recognition based on deep convolutional neural networks," *Jordanian Journal of Computers and Information Technology (JJCIT)*, vol. 3, no. 3, pp. 186–200, 2017.
[12] A. Lawgali, M. Angelova, and A. Bouridane, "HACDB: Handwritten Arabic characters database for automatic character recognition," *Proc. IEEE EUVIP*, Paris, France, June 2013, PP. 255-259.
[13] S. Mozaffari, K. Faez, F. Faradji, M. Ziaratban and S.M. Golzan, "A comprehensive isolated Farsi/Arabic character database for handwritten OCR research," *Proc. IWFHR*, La Baule, France, Oct. 2006, pp 385-389.
[14] C.-L. Liu and C.Y. Suen, "A new benchmark on the recognition of handwritten bangla and farsi numeral characters," *Pattern Recognition*, vol. 42, no. 12, Dec. 2009, pp. 3287–3295.
[15] D. Sudha and M. Ramakrishna, "Comparative study of features fusion techniques," in *2017*

International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT). IEEE, 2017, pp. 235–239.

[16] B. Fernando, E. Fromont, D. Muselet, and M. Sebban, "Discriminative feature fusion for image classification." *Proc. IEEE CVPR*, Rhode Island, USA, June 2012, pp. 3434-3441.

[17] Y. Alginahi, *Preprocessing techniques in character recognition*, INTECH Open Access Publisher, 2010.

[18] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, Jan. 1979, pp. 62-66.

[19] M. G. Sarowar, A. A. Jamal, A. Saha, and A. Saha, "Performance evaluation of feature extraction and dimensionality reduction techniques on various machine learning classifiers," in *2019 IEEE 9th International Conference on Advanced Computing (IACC)*. IEEE, 2019, pp. 19–24.

[20] P. Antonik, N. Marsal, D. Brunner, and D. Rontani, "Comparison of feature extraction techniques for handwritten digit recognition with a photonic reservoir computer," in *International Conference on Artificial Neural Networks*. Springer, 2019, pp. 175–179.

[21] J.A. Rodrıguez, and F. Perronnin, "Local gradient histogram features for word spotting in unconstrained handwritten documents," *Proc. 1st ICFHR*, 2008, pp. 7-12.

[22] N. Suguna, and K. Thanushkodi, "An improved K-nearest neighbor classification using Genetic Algorithm," *International Journal of Computer Science Issues*, vol. 7, no. 2, 18-21.

[23] J. Gou, W. Qiu, Z. Yi, X. Shen, Y. Zhan, and W. Ou, "Locality constrained representation-based k-nearest neighbor classification," *Knowledge-Based Systems*, vol. 167, pp. 38–52, 2019.

[24] C.J.C. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, June 1998, pp. 121-167.

[25] J. S. Paiva, J. Cardoso, and T. Pereira, "Supervised learning methods for pathological arterial pulse wave differentiation: a svm and neural networks approach," *International journal of medical informatics*, vol. 109, pp. 30–38, 2018.

[26] V. Passricha, and R. K. Aggarwal, "Convolutional support vector machines for speech recognition," International Journal of Speech Technology, 2018, pp.1-9.

[27]P. Wang, and C. Xu, "Robust Face Recognition via Sparse Representation".

[28] H.A. Khan, A.A. Helal, K.I. Ahmed, "Handwritten bangla digit recognition using sparse representation classifier," Proc. IEEE ICIEV, Dhaka, Bangladesh, May 2014, pp. 1–6.

[29] D.L. Donoho, "For most large underdetermined systems of linear equations the minimal l1norm solution is also the sparsest solution," *Communications on Pure and Applied Mathematics*, vol. 59, no. 6, June 2006, pp. 797–829..

[30] Al-Ohali, Yousef, M. Cheriet, and C. Suen, "Databases for recognition of handwritten Arabic cheques," Pattern Recognition, vol. 36, no. 1, Jan. 2003, pp. 111-121.

[31] L. Souici, N. Farah, T. Sari, and M. Sellami, "Rule based neural networks construction for handwritten arabic city-names recognition," AIMSA, Aug. 2004, pp. 331-340.

[32] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," ACM Transactions on Intelligent Systems and Technology, vol. 2, no.3, Apr. 2011, p. 27.

[33] G. Madzarov, D. Gjorgjevikj, and I. Chorbev, "A multi class SVM classifier utilizing binary decision tree," *Informatica*, vol. 33, no. 2, May 2009, pp. 233-241.