Comparison between Feature Based and Deep Learning Recognition Systems for Handwriting Arabic Numbers

Ali Kamal Taqi , Ayad E. Korial
Computer Engineering Department, University of Technology, Iraq

Abstract

Feature extraction from images is an essential part of the recognition system. Calculating the appropriate features is critical to the part of the classification process. However, there are no standard features nor a widely accepted feature set exist applied to all applications, features must be application dependent. In contrast, deep learning extract features from an image without need for human hard-coding the features extraction process. This can be very useful to build a model for classification which can classify any type of images after trained with enough images with labels then the trained model can be used in different recognition applications to classify. This paper presents two techniques to build recognition system for Arabic handwriting numbers, the feature-based method shows accepted results. However, the deep learning method gives more accurate results and required less study on how Arabic number is written and no hand-coding algorithms needed for feature extraction to be used in the classification process.

Keywords: Handwriting Recognition, Image Processing, Features Extraction, Machine Learning, Deep Learning, Classification.

1. Introduction

The Eastern Arabic numerals also known as Arabic Eastern numerals and Arabic–Hindu numerals, are the symbols used to express the Hindu–Arabic numeral system, in association with the Arabic alphabet in the countries of the Arab world, the Arabian Peninsula, and its variant in other countries that use the Perso-Arabic script in Asia and the Iranian plateau.

The numeral system introduces from an ancient Indian numeral system, which was used by engineer Khwarazmi and the medieval-era Iranian mathematician in the book On the Calculation with Hindu Numerals, whose name was Latinized as Algoritmi.

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

Table (1): Hindu-Arabic numerals

This paper present comparison between two type of recognition system the first system is feature based system while the second one depends on deep learning; each system shows different workflow and different result accuracy results.

Saeeda et al. [1] presented a review on numeral recognition for Urdu, Arabic, and Farsi. Most of the published paper for the Urdu language since 2003 and recent publications in the Arabic and Farsi languages have been summarized in this survey paper. The Unicode system and writing glyph of digits have been analyzed in Arabic language and its derivative languages.

Yasser, Member, and IACSIT [2] proposed an Arabic LP recognition, this system was recognized both Arabic and Indian numerals as well as limited Arabic and Latin alphabets. The system used many preprocessing steps in order to produce segmented characters of the LPs images. The feature extraction step used to count of black pixels from the horizontal projection profiles in addition to the black pixel distributions in divided zones of the character image. The recognition step used both a distance classifier and Neural Network (NN) classifier to discriminate between the different characters.

Amr et al. [3] Using Morphological operations introduced an Automatic Number Plate Recognition System (ANPR), Edge Detection Techniques and Histogram manipulation for characters segmentation and plate localization. Artificial Neural Networks are used for character classification and recognition.

Rawan, *Muayad, and Ayad* [4] proposed a mobile application for human eye tracking using the front camera run in real time based on OpenCV (2.4.9) library. the proposed system does not need any calibration with additional sensors, in which the accuracy and efficiency are guaranteed.

Ahmed and Alaa [5] presented a license plate recognition system for the Egyptian plates introduced in 2008. The proposed system composed of three main stages; localization & skew correction stage, segmentation stage and recognition stage. In the first stage, find the plate candidates in the image and to measure the skew angle. The second stage, find objects belong to license number. The final stage, used adapt template match technique to recognize the digits and letter groups separately after normalizing them. The system accuracy was 81% and average time per frame was 24 msec/frame.

Cyrus [6] suggested handwriting engine to recognize Arabic letters. This was achieved by collecting writing data from Arabs and using present utilities to process the data by clustering it using divisive hierarchical clustering. Some features were attached to the existing recognition engine to eliminate ambiguity in problematic character pairs. The attained recognition rate was 95%.

Rawan et al. [7] build a classifier that can easily recognize offline handwritten Arabic. The proposed algorithm was developed using MATLAB and tested with a large sample of handwritten numeral datasets for different writers in different ages. Pattern recognition techniques are used to identify Hindi (Arabic) handwritten numerals. The recognition rates were achieved in the range from 95% up to 99%.

Muayad, Saja and Ali [8] proposed an algorithm for edge detection of dental x-ray images by the process the image and extract a feature of investigated teeth using image processing model and segmentation the obtained result shows a satisfied accuracy of edge detection for the interested object.

Loay and *Faisel* [9] developed a heuristic-based method for recognizing numeral free handwritten objects. The introduced method for extracting features from patterns based on the percentage of strokes in both horizontal and vertical directions and some morphological operations. The recognition rate was 98.15%, the number of tested samples was 4500 samples.

H.A. Jeiad [10] proposed an Indian number handwriting recognition model (INHRM) to extract features and used multiclass SVM (MSVM) approach in the training and testing processes, The results showed that the proposed model achieved a relatively high percentage of exactness of around 97%.

Mohammed and *Rolla* [11] proposed Alphanumeric VGG net for Arabic handwritten alphanumeric character recognition. Alphanumeric VGG net was constructed by thirteen convolutional layers, two max-pooling layers, and three fully-connected layers. The accuracy was 99.66% for the ADBase database and 97.32% for the HACDB database.

Hassan and Ali [12] Present an isolated image points feature based technique used to find the implicit information in the face image and to extract features to be used in the neural network for recognition using a neural network.

Fahad et al. [13] presented an approach extract Arabic numerals from digital images to be used in an automatic recognition system. By normalizing the input image and pre-processed it to an appropriate form. Different numbers represented by segmenting the words into individual objects. An expert system based on a set of if-else rules used, where each number represented by a set of rule. Finally, precise tests were taken on 226 random Arabic numerals selected from 40 images of Iraqi car plate numbers. The accuracy was 97%. In the feature-based section, an algorithm is proposed to extract the feature which include number of starting and intersection points, average zone values and chain vector which indicate hand movement when number is written using the neighbor pixel direction, the feature vector implemented in different classification technique includes: neural network, support vector machine, decision tree, nearest neighbor classifiers, ensemble classifiers. In the deep learning section, the images fed to the deep neural network without any image preprocessing or any image enhancement technique, using inception module which maximizes information flow into the network and increasing the number of features by increasing the depth, using pooling layer and softmax as the final classifier.

2. Feature Based Recognition

The model of the feature based system with classic stages. The algorithm consists of four stages begins with image acquisition and ends with the result class.

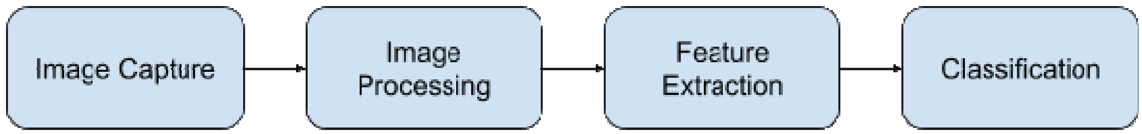

Figure (1). Block diagram of feature based algorithm

2.1 Image Capture

handwriting Images were collected from 60 students, makes 600 the total number of samples, each number have 100 samples. The images scanned with 300 dpi to obtain convenient image quality and input to the model with 90x90 dimension to have uniform image set of the same size to be processed in the next stage

2.2 Image Preprocessing

The preprocessing stage consist of image enhancement technique which include equalize the brightness level using histogram equalization and improving the local contrast and edges using contrast limited adaptive histogram equalization then convert the image from grayscale to binary using Otsu's method and using morphology of opening and closing to fill the gaps and remove noise then resize the image to only area of interest using static image zoning as shown on figure (2).

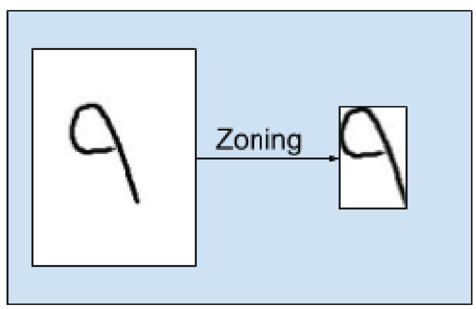

Figure (2): Static image zoning

Finally using skeleton (thinning) operation to get a one-pixel wide object to prepare the image for feature extraction. The image processing stages can be described in the following flowchart

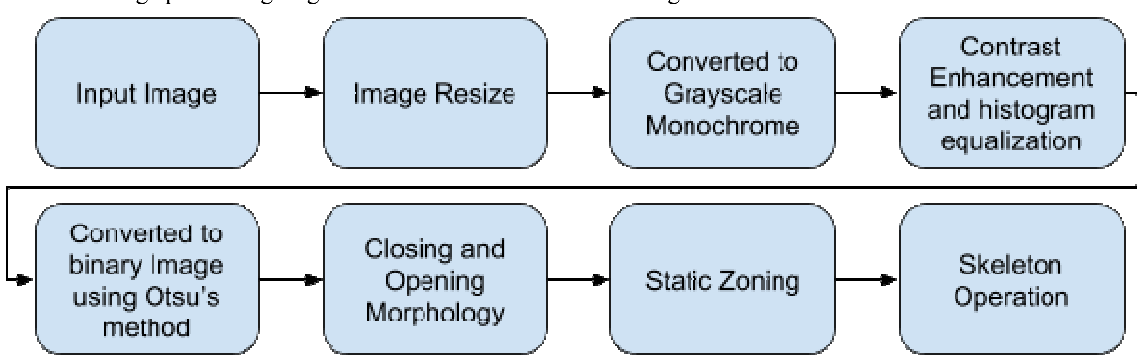

Figure (3): The image processing stages

2.3 Features Extraction

Calculating features is an essential part of the classification process. However, there are no standard features nor a widely accepted features set exist applied to all applicants, the feature must be application dependent. In general, features divided into the structural features and global transformation features, each has a different type of features as shown in figure(4)

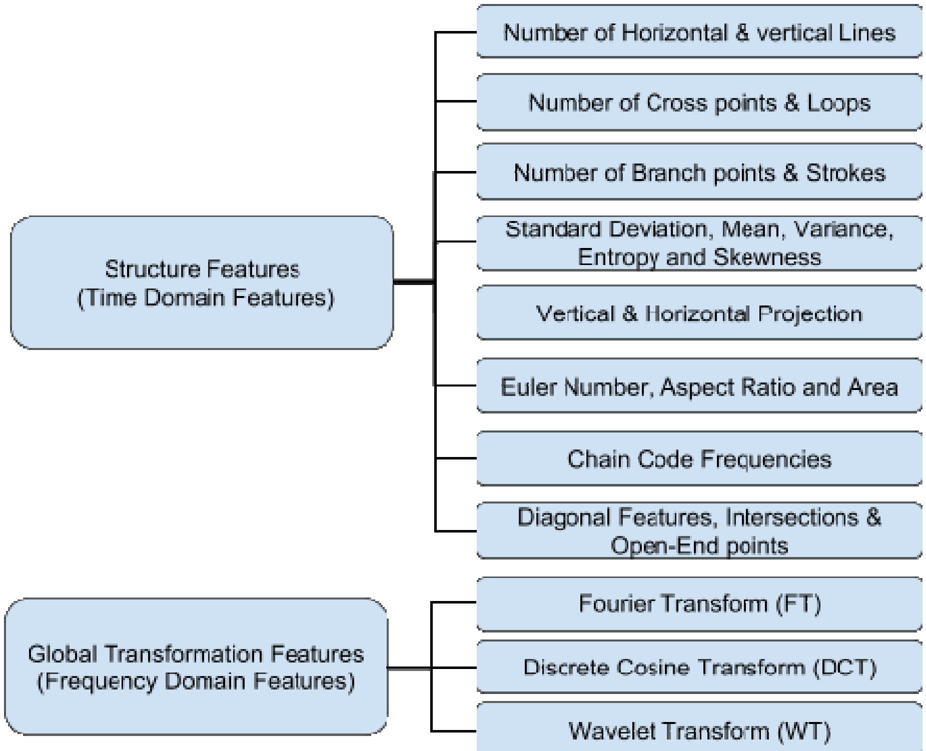

Figure (4): Features types

The proposed algorithm for this feature extraction includes calculate the means and finding the starting and intersection points and partition the image to four regions to calculate the average of each region to get vector of four average region zone then detecting connected edge based on 8-connectivity to construct the chain vector which contain the path of the image object depending on the pixels neighbor then calculate the frequency and normalized the chain vector to size on 10 and check if the normalized vector need padding. then plot the result of the chain vector to get the output of the digitized image.

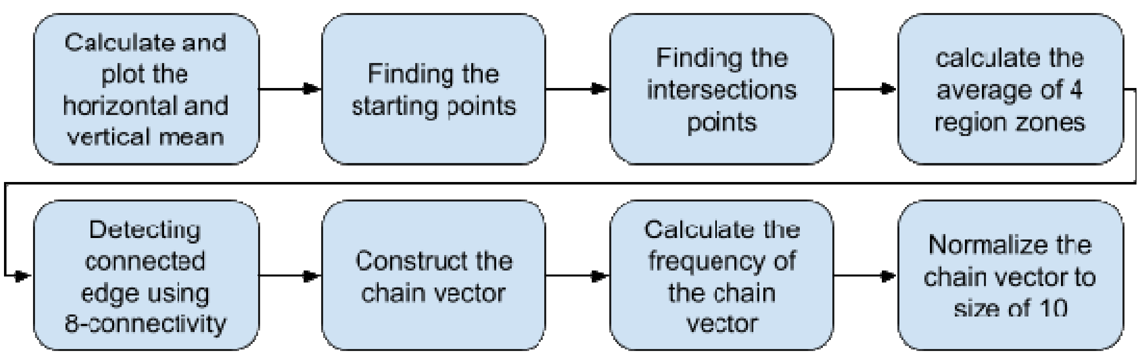

Figure (5): Feature extraction process

Horizontal and vertical projection means calculated from the mean of pixels' value at each dimension. These projections are computed as

$$H[i] = \frac{\sum_{j=0}^{m-1} I[i,j]}{m} \qquad V[j] = \frac{\sum_{j=0}^{n-1} I[i,j]}{n}$$

Where the I[i,j] is an image with *m* row and *n* column, H[i] is the projection of a binary image along the rows (horizontal projection). The V[i] is the projection of the columns (vertical projection).

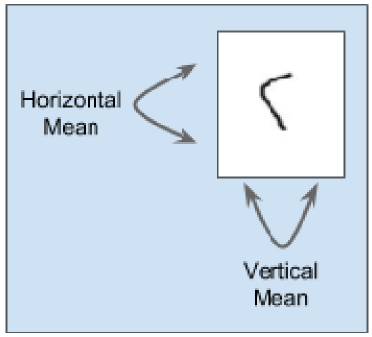

Figure (6): Horizontal and vertical projection means

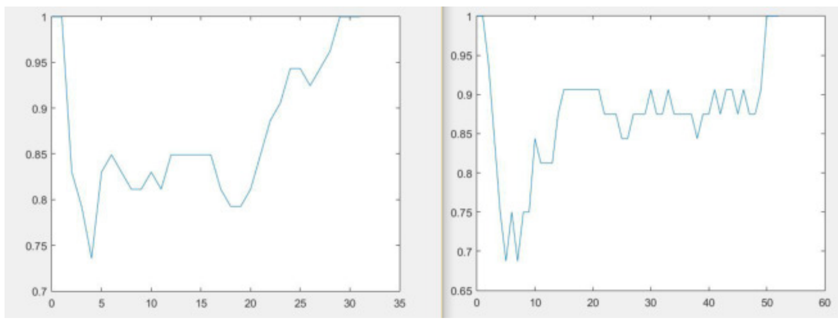

Figure (7): Horizontal and vertical projection means for number 2 in Arabic

The 8-connectivity pixels are neighbors to all pixel that reaches one of their corners or edges. These pixels are attached vertically, horizontally, and diagonally.

For pixel A with coordinate B is considered neighbor to pixel A if its coordinate is

(x-1,y+1)	(x,y+1)	(x+1,y+1)
(x-1,y)	(x,y)	(x+1,y)
(x-1,y-1)	(x,y-1)	(x+1,y-1)

Figure (8): The 8-connectivity pixels are neighbors

The starting points are the number and location of the object where the image pixel has only one neighbor

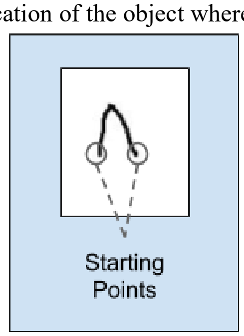

Figure (9): Starting points

The intersection points are the number and location of the object where the image pixel has more than one neighbor

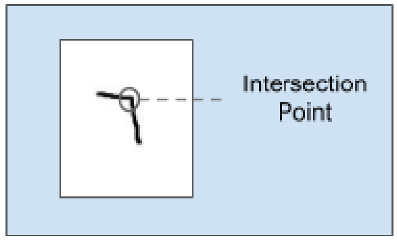

Figure (10): Intersection points

To calculate the average zone image is partitioned into four zones and calculate the average pixel values in each region, these projections are computed as

$$\boldsymbol{R}[1] = \frac{\sum_{i=0}^{(m/2)-1}\sum_{j=0}^{(n/2)-1}\boldsymbol{I}[i,j]}{(m/2)+(n/2)} \qquad \boldsymbol{R}[2] = \frac{\sum_{i=(m/2)}^{m-1}\sum_{j=0}^{(n/2)-1}\boldsymbol{I}[i,j]}{(m/2)+(n/2)}$$

$$\boldsymbol{R}[3] = \frac{\sum_{i=0}^{(m/2)-1}\sum_{j=(n/2)}^{n-1}\boldsymbol{I}[i,j]}{(m/2)+(n/2)} \qquad \boldsymbol{R}[4] = \frac{\sum_{i=(m/2)}^{m-1}\sum_{j=(n/2)}^{n-1}\boldsymbol{I}[i,j]}{(m/2)+(n/2)}$$

Where is the image with *m* row and *n* column, R is the average of the zone region.

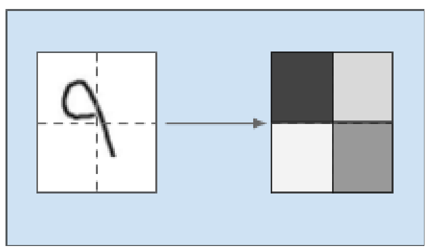

Figure (11): average of the zone region

The chain vector computed using one of the previously obtained starting points, search using 8-connectivity neighbors to find and store neighbor location, then move toward the next neighbor, if intersection point found then store the intersection coordinate and continue toward the direction of the line then when reach the end return to the intersection point to visit the other line using this method solve the problem of falling into the local minima which can occur when there is an intersection and the chain vector must choose the path direction.

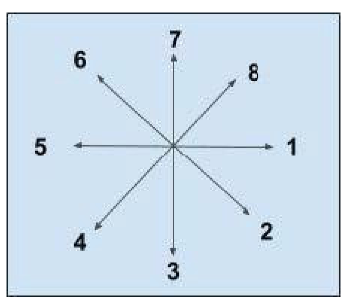

Figure (12): Chain vector direction values

This example demonstrates how the chain vector stage obtained from an ideal image object

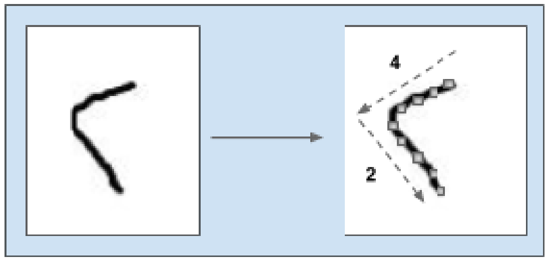

Figure (13): Constructing the chain vector

The chain vector depends on the handwriting movements. The calculation of the chain vector begins with by moving toward the part before the intersection after completion it will return to visit the other line segments. After obtaining the chain vector it is possible to reconstruct the object depending on one of the starting points and the chain as following

Chain_Vector = [4 4 4 4 2 2 2 2 2 4 4 4 4 2 2 2 2 2]

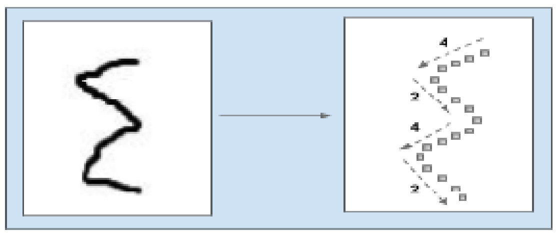

Figure (16): simplified example of chain vector construction

When there is an intersection point using the previously mentioned method can solve the issue and obtain complete chain vector as the following figure (14).

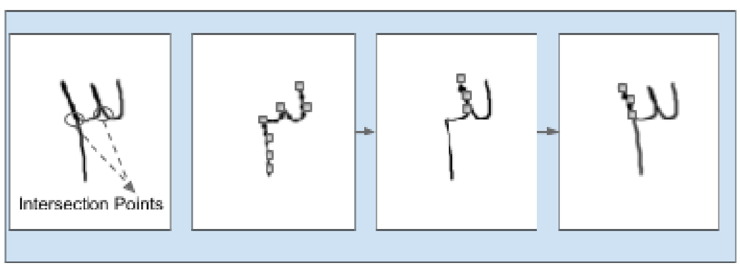

Figure (14): Constructing the chain vector with many intersection points

The result reconstructed chain vector of the digitized image shown in the figure (15).

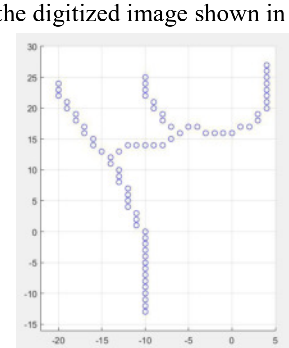

Figure (15): reconstructed number three in Arabic from the obtained chain vector

Computer Engineering and Intelligent Systems
ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)
Vol.9, No.4, 2018

The output of the feature extraction is feature vector contains the following

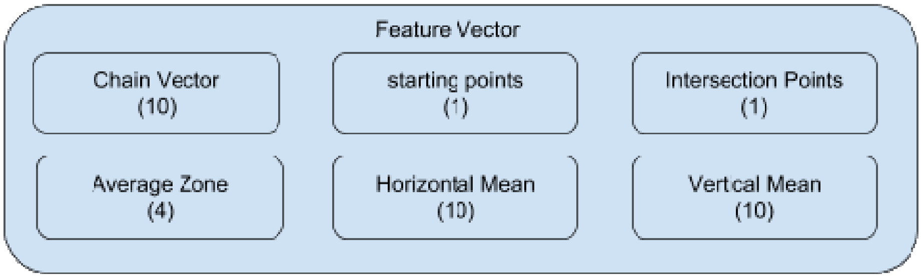

Figure (17): Feature vector values and its sizes

Total vector size for each image is 36 elements, however, in the implementation stage, the horizontal and vertical mean were eliminated to enhance the computational speed and simplify the classification computation.
Normalization is used to remove the redundancy of the chain vector and at the same time preserve the features of the chain vector resulting essential neighbor points directions and therefore reduce the feature vector size, this form of implementing simple dimension reduction technique can improve the feature vector and speed calculations. trial and error used and find that a vector of size ten is suitable for distinguishing different objects.
The frequency of the neighbors is calculated Then ignore the neighbors with small frequency regarding others in order to avoid the small and uncorrelated movement, to calculate the normalizes the vector to the size of ten as following procedure

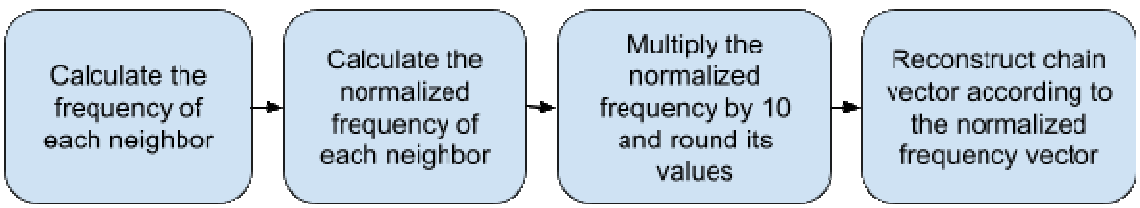

Figure (18): chain vector normalization process

normalized frequency calculated using the following equation

$$NF = \frac{fk}{\sum F}$$

Where NF is the normalized frequency, f is the frequency of the neighbor k and F is the frequency of all neighbors

$$NV = Round(NF * 10)$$

Where NV is the normalized vector
The following example (number 4 in Arabic) demonstrate the procedure of finding the normalized chain vector

N	**5**	**4**	**3**	**2**	**1**	**2**	**4**	**5**	**4**	**3**	**2**	**1**
F	**3**	**6**	**4**	**2**	**10**	**2**	**2**	**5**	**4**	**8**	**4**	**6**
NF	**0.054**	**0.107**	**0.071**	**0.036**	**0.179**	**0.0036**	**0.036**	**0.089**	**0.071**	**0.142**	**0.071**	**0.107**
NV	**0.54**	**1.107**	**0.71**	**0.36**	**1.79**	**0.036**	**0.36**	**0.89**	**0.71**	**1.142**	**0.71**	**1.07**
	1	**1**	**1**	**0**	**2**	**0**	**0**	**1**	**1**	**1**	**1**	**1**

Table (2): Normalized chain vector process
Where N is the neighbor pixel, F is the frequency, NF is the normalized frequency, NV is the normalized vector.
Padding is used when applying the normalization equation may give the size of 9 in some cases

2.4 Classification

The classification process has 10 class from number 0 to 9 in Arabic. The training set contains 600 samples each class has 60 samples, the training process using 80% of the samples and the remaining 20% used for the testing process.

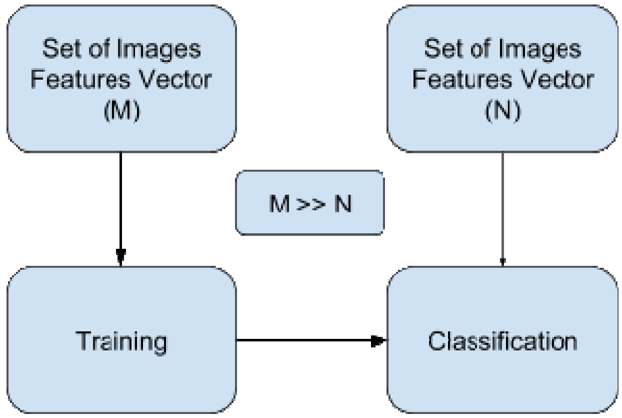

Figure (19): General classification model

The feature vector size is 600x16 consist of 600 sample image each the sample have 16 features. These sample images were collected from different students

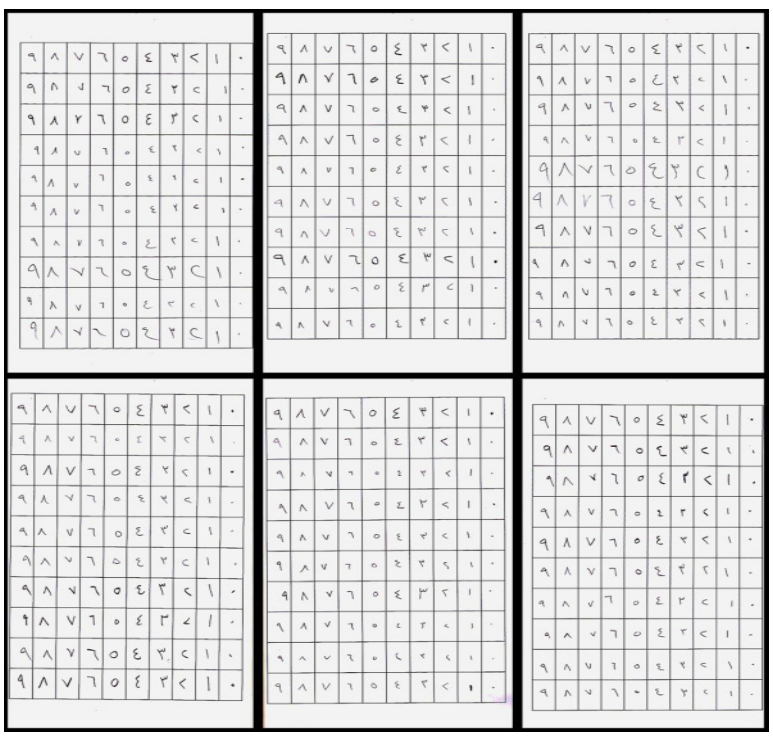

Figure (20): The 600 samples for Arabic handwriting numbers

Different type of classification used for the feature-based method

1. Neural Network Pattern Recognition: neural network used to classify the input feature vector into set of target categories, a two-layer feed-forward network with sigmoid Hiffen and SoftMax output neurons, given enough neurons in its hidden layer, the network trained with scaled conjugate gradient backpropagation, with 10 neutrons used in the hidden layer of 16 input and 10 output.

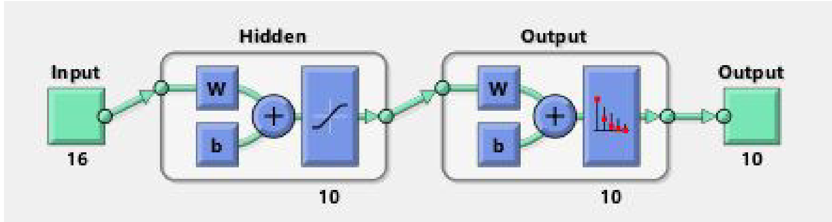

Figure (21): Neural network

layers the samples divided into the kind of samples

a. Training: these are presented to the network during training and the network is adjusted according to its error, 80% of feature vector used for training
b. Validation: these are used to measure network generalization, and to halt training when generalization stops improving, 10% of feature vector used for Validation
c. Testing: provide an independent measure of network performance and have no effect on training.

After training, 10% of feature vector used for testing, the results of minimizing cross-entropy and error percentage shown in table (3).

Stage	Samples	cross-entropy	error percent
Training	480	1.30100	16.6666
Validation	60	5.58395	21.6666
Testing	60	5.60837	31.6666

Table (3): Neural network stages cross-entropy and error percent

The neural network gives accuracy of 81.57% with the following confusion matrix

OUTPUT CLASS	**0**	**49**	**2**	**2**	**1**	**4**	**1**	**0**	**1**	**1**	**0**	**80.3%**
	1	**4**	**55**	**0**	**0**	**1**	**2**	**0**	**3**	**2**	**0**	**82.1%**
	2	**1**	**0**	**50**	**2**	**9**	**2**	**0**	**1**	**1**	**3**	**72.5%**
	3	**1**	**0**	**1**	**50**	**0**	**0**	**3**	**2**	**0**	**8**	**76.9%**
	4	**1**	**0**	**6**	**0**	**43**	**0**	**1**	**0**	**1**	**1**	**81.1%**
	5	**0**	**1**	**0**	**0**	**0**	**55**	**0**	**1**	**0**	**1**	**94.8%**
	6	**0**	**0**	**0**	**4**	**0**	**0**	**53**	**2**	**1**	**2**	**85.5%**
	7	**0**	**1**	**0**	**1**	**1**	**0**	**0**	**43**	**6**	**0**	**82.7%**
	8	**2**	**1**	**0**	**0**	**2**	**0**	**0**	**6**	**48**	**3**	**77.4%**
	9	**2**	**0**	**1**	**2**	**0**	**0**	**3**	**1**	**0**	**42**	**82.4%**
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	
Target Class												

Table (4): Neural network confusion matrix

Neural network performance and Receiver operating characteristic is shown in the figure (22)

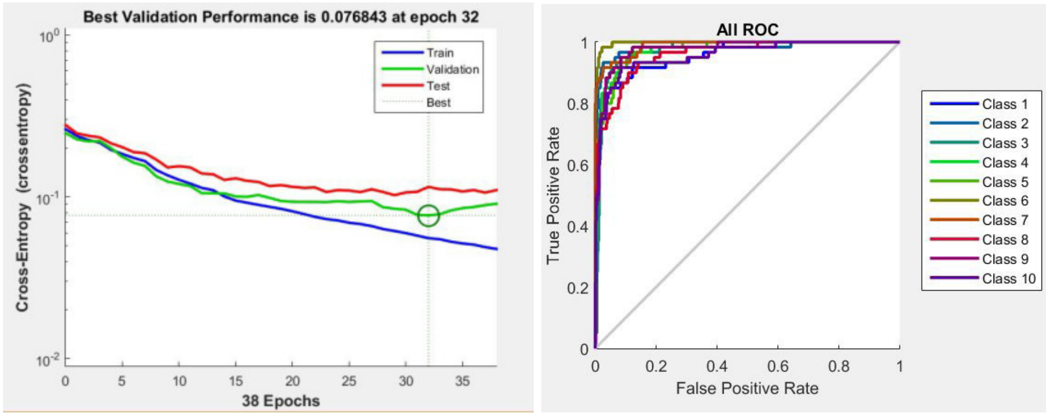

Figure (22): Nerul network cross-entropy and ROC

Neural network training error histogram is shown in figure (23).

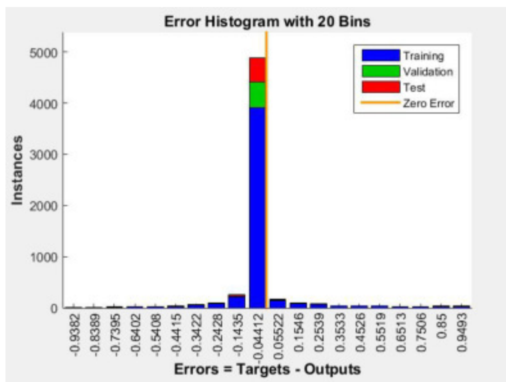

Figure (23): Neural network error histogram

2. Support Vector Machine: Support vector machines have high predictive accuracy, medium fitting speed, and can have good prediction speed and memory usage with few support vectors. Multi-class SVM used (One-vs-All) with cubic kernel function gives an accuracy of 85.0% with the following confusion matrix

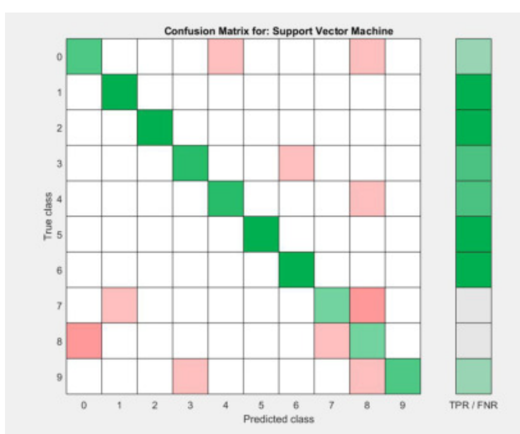

Figure (24): SVM confusion matrix

3. Decision Tree: are easy to interpret, fast for fitting and prediction, and low on memory usage, but they can have low predictive accuracy. The Tree used with 100 maximum number of

splits and twoing rule used as the split criterion gives an accuracy of 76.7% with the following confusion matrix.

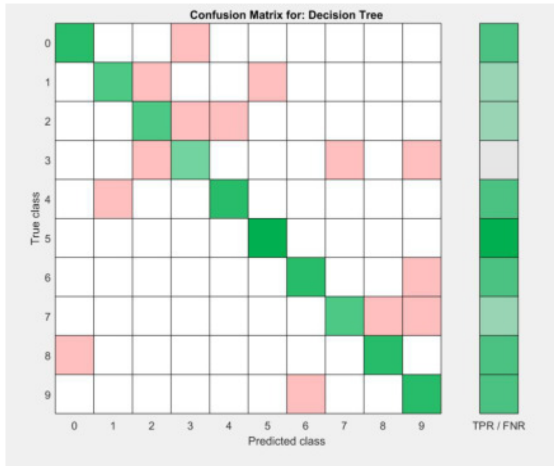

Figure (25): Decision Tree confusion matrix

4. Nearest Neighbor Classifiers: has good predictive accuracy in low dimensions, but not in high dimensions. They have fast fitting speed, medium prediction speed, high memory usage, and are not easy to interpret. Weighted KNN with 10 neighbors and city block as distance metric gives an accuracy of 81.7% with the following confusion matrix

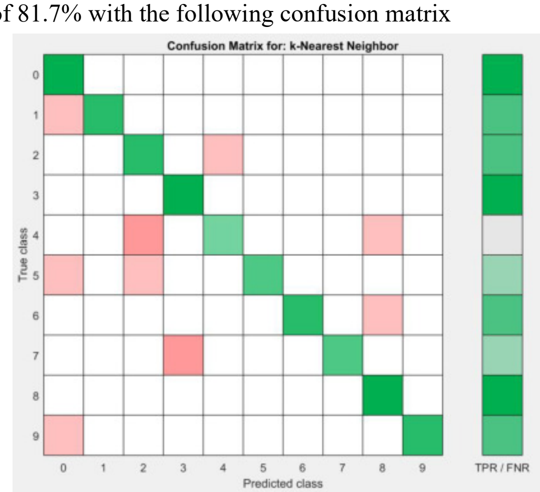

Figure (26): Nearest neighbor confusion matrix

5. Ensemble Classifiers: meld results from many weak learners into one high-quality ensemble predictor. Ensemble with Bag method, decision tree as learner type, 100 number of the learner with 0.1 learning rate gives an accuracy of 91.7% with the following confusion matrix

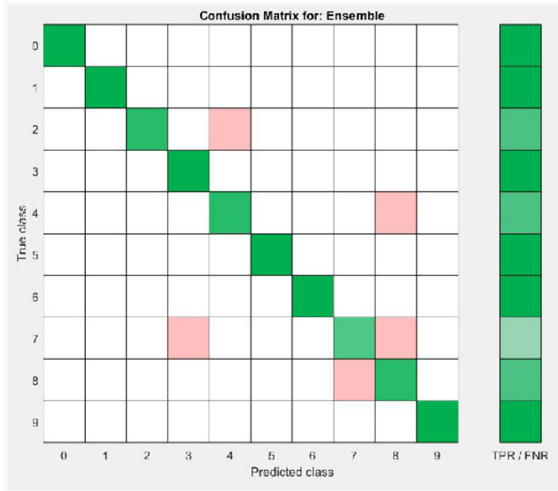

Figure (27): Ensemble classifier confusion matrix

3. Deep Learning based Recognition

Deep Learning are powerful and popular algorithms. New version of the Inception v3 modules released in December 2015. Google's freely available this model and it is a great recognition model for objects. A list of the original ideas is:

-Constructing networks that balance depth and width to maximize information flow into the network (increase the feature maps before each pooling).

-The width of the layer or number of features increased systematically when depth is increased.

-Increase the combination of features before next layer by increase width at each layer.

-When possible, only use 3×3 convolution, decomposed filter of 5×5 and 7×7 with multiple 3×3. See figure (28):

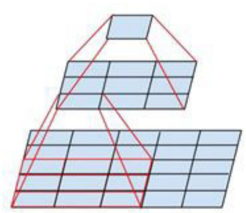

Figure (28): filter 5x5 to 3x3

matrix the new inception module thus becomes:

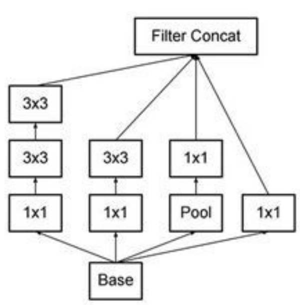

Figure (29): Inception Model

Decomposed filters by flattened convolutions into more complex modules:

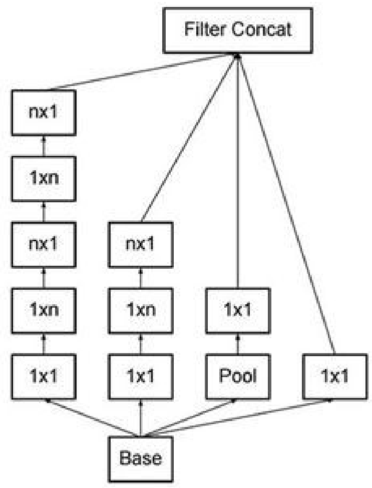

Figure (30): Flattened convolution inception model

inception modules can also decrease the size of the data by providing pooling while performing the inception computation. This is basically identical to performing a convolution with strides in parallel with a simple pooling layer:

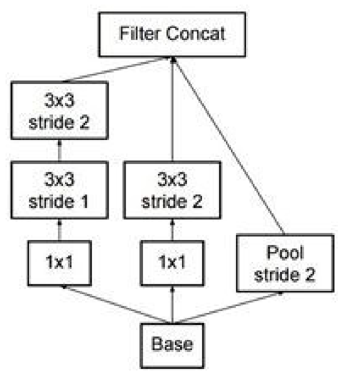

Figure (31):

Train a new top layer that can recognize other classes of images and learn with Inception v3. We train a SoftMax layer on top of this representation. Inception v3 uses a pooling layer and SoftMax as the final classifier, assuming the SoftMax layer contains 10 labels. Assumes have a folder containing subfolders, each full of images for each label. The folder dataset should have a structure as shown in figure 32.

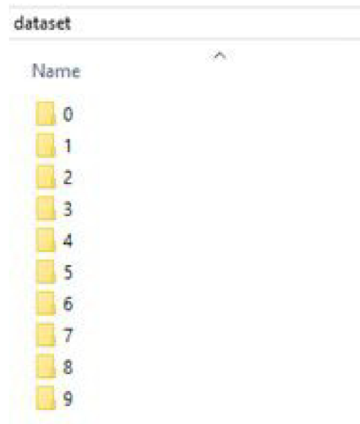

Figure (32): folder structure of dataset

Each of these subfolders contains 60 images so for total 600 images. The subfolder names are important since they define what label is applied to each image, but the filenames themselves don't matter. This produces a new model file that can be loaded and run by any TensorFlow program. The overall stages are shown in figure (33).

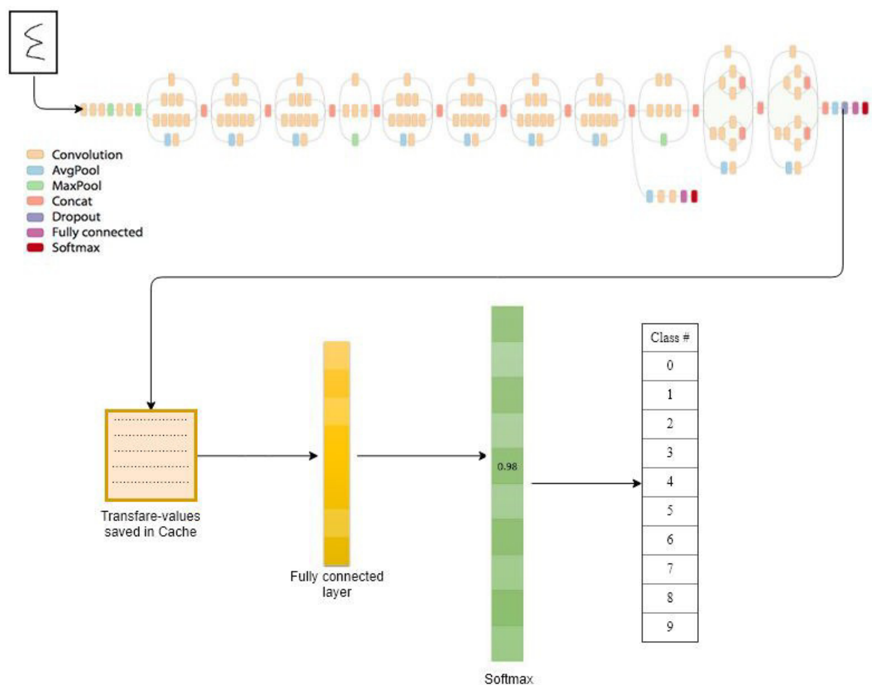

Figure (33): Overall block diagram for deep learning model

The deep neural network gives an accuracy of 93.2%. The result Train accuracy, validation accuracy and cross entropy shown in figure (34).

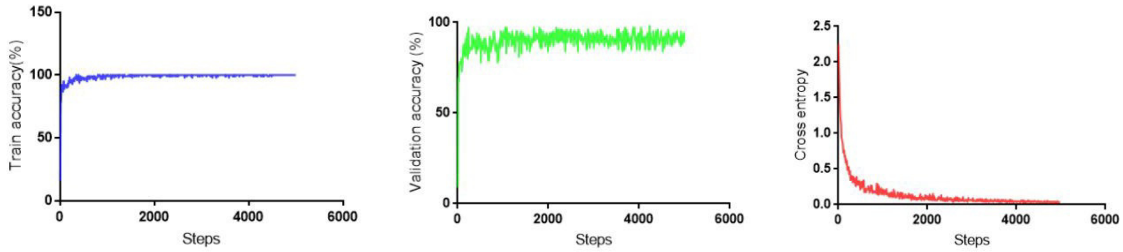

Figure (34): Train accuracy, validation accuracy and cross entropy

4. Conclusion

This paper present comparison between feature based and deep learning recognition system for Arabic handwriting number with 600 sample images. In the feature based section, an algorithm is proposed to extract the feature which include number of starting and intersection points, average zone values and chain vector which indicate hand movement when number is written using the neighbor pixel direction, the feature vector implemented in different classification technique includes: neural network gives accuracy of 81.57%, support vector machine gives accuracy of 85.0%, decision tree gives accuracy of 76.7 %, nearest neighbor classifiers gives accuracy of 81.7%, ensemble classifiers gives accuracy of 91.7% which is the highest accuracy reached in the feature based system. In the deep learning section, the images faded to the deep neural network without any image preprocessing or any image enhancement technique, using inception module which maximize information flow into the network and increasing the number of features by increasing the depth, using pooling layer and softmax as final classifier the system gives accuracy of 93.2% .

5. References

[1] Naz S, Ahmed SB, Ahmad R, Razzak MI. Arabic Script based Digit Recognition Systems. In International Conference on Recent Advances in Computer Systems (RACS) 2016 (pp. 67-73).

[2] Alginahi YM. Automatic arabic license plate recognition. International Journal of Computer and Electrical Engineering. 2011 Jun 1;3(3):454.

[3] Badr A, Abdelwahab MM, Thabet AM, Abdelsadek AM. Automatic number plate recognition system. Annals of the University of Craiova-Mathematics and Computer Science Series. 2011 Mar 15;38(1):62-71.

[4] Rawan Ali Taban, Dr. Muayad Sadik Croock and Ayad E. Korial, "Eye Tracking Based Mobile Application", International Journal of Advanced Research in Computer Engineering & Technology, 2018.

[5] El Rahman AA, Hamdy A, Zaki F. Automatic arabic number plate recognition. In Electronics, Communications and Computers (JEC-ECC), 2013 Japan-Egypt International Conference on 2013 Dec 17 (pp. 23-28). IEEE.

[6] Bakhtiari-Haftlang C. Arabic online handwriting recognition. Lund University; 2008.

[7] Zaghloul RI, Enas DM, AlRawashdeh F. Recognition of Hindi (Arabic) handwritten numerals. American Journal of Engineering and Applied Sciences. 2012;5(2):132-5.

[8] Croock, Muayad Sadik, Saja Dheyaa Khudhur, and Ali Kamal Taqi. "Edge Detection and Features Extraction for Dental X-Ray." Eng. &Tech. Journal 34; 2016.

[9] Ghaleb MH, George LE, Mohammed FG. Numeral handwritten Hindi/Arabic numeric recognition method. International Journal of Scientific & Engineering Research. 2013 Jan;4(1).

[10] H.A. Jeiad, "Indian Number Handwriting Features Extraction and Classification using Multi-Class SVM", Engineering and Technology Journal, Vol. 36, No. 1, pp. 33-40, 2018

[11] Mudhsh M, Almodfer R. Arabic Handwritten Alphanumeric Character Recognition Using Very Deep Neural Network. Information. 2017 Aug 31;8(3):105.

[12] Hassan, Hassan Jaleel, and Ali Kamal Taqi. "An Algorithm for Face Recognition based on Isolated Image Points with Neural Network." International Journal of Computer Applications 150.2 (2016).

[13] Malallah FL, Saeed MG, Aziz MM, Arigbabu OA, Ahmad SM. Off-Line Arabic (Indian) Numbers Recognition Using Expert System. International Journal of Advanced Computer Science and Applications. 2016 Apr 1;7(4):397-406.