Journal of Theoretical and Applied Information Technology

15th September 2023. Vol.101. No 17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN OCR FOR ARABIC CHARACTERS WITH ADVANCED BASELINE SEGMENTATION AND ENHANCED CONVOLUTIONAL NEURAL NETWORK FOR CLASSIFICATION

ASHIQ V M 1 , DR. E. J. THOMSON FREDRIK 2

1 Research scholar, Department of Computer Technology, Karpagam Academy of Higher Education, Coimbatore , India

2 Professor, Department of Computer Applications, Karpagam Academy of Higher Education, Coimbatore, India

E-mail: 1 vmashiq@gmail.com, 2 thomson500@gmail.com

ABSTRACT

Human-computer communication is referred to as "Natural Language Processing (NLP)". There are several "Data Mining" along with "Machine Learning" techniques available, and the NLP goal is to allow machines to produce meaningful knowledge from natural language input. The "Arabic Optical Character Recognition (AOCR)" is a continuing process due to the obvious complex pattern and syntactic of Arabic words. Scanned documents and printed text recognition have been a major focus of AOCR research in the last several years. However, the outcomes of the AOCR study are unsatisfactory, and more work has to be done in this area. The segmentation process of AOCR is crucial for the recognition of specific Arabic texts. The character element would have a distinct representation if the fundamental form of an Arabic character is incorrectly segmented. For the segmentation of Arabic letters, we propose a "Baseline Segmentation (BS)" in this research. Every link between consecutive Arabic letters may be found along the "Base-line", which is an Arabic word's "Medium-line". Though an attempt to tackle the issues of recognizing digital Arabic Characters, including solitary numbers, characters, and vocabulary, in this research we develop a new model leveraging the "Convolutional Neural Network (CNN)" after the segmentation procedure has been completed. In particular, we enhance the conventional CNN (ECNN) model by using "Batch Normalization" and "Dropout Regularization" parameters to retrieve features that are optimum contextually. Preventing overfitting while also improving generalization is the goal of this approach. The suggested ECNN framework is designed using a multitude of convolutional layers. Multiple evaluation criteria, such as "Accuracy", "Precision", and "Recall", are used to thoroughly analyze the developed ECNN model. Further, we conduct experiments on a dataset obtained and compare the results to those obtained using preexisting EKNN and FKNN methods. The proposed ECNN approach achieves more accuracy than either the EKNN or FKNN approaches.

Keywords: *NLP, AOCR, Baseline Segmentation, ECNN, EKNN, FKNN*

1. INTRODUCTION

The term "Optical Character Recognition" (OCR) refers to the procedure of digitizing text from a graphic image into a machine-readable structure. Due to this, this might leverage a wide range of computing scientific research specializations, such as image analysis, pattern classification, NLP, cognitive computing, and distributed databases [1].

The overall purpose of establishing an OCR technology that possesses comparable word recognition as individuals has not yet been reached, particularly in the context of the Arabic script, following much research [2].

The OCR had already captivated the attention of academics for several reasons, including the fact that it enhances human-machine interaction in a wide variety of contexts and the fact that it is a highly difficult issue to solve in an attempt to settle the shortfall in computer and user reading skills. Business context, verifying cheques, and other financial, commercial, and data input operations are just a few examples [3].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Many existing OCR software are designed to read written English rather than Arabic since the former has more easily distinguishable characters according to the usage of spaces between words, while the latter has more inherent difficulties due to the use of Arabic script. Because of this, tools for recognizing the word in English are more refined and user-friendly than those for other languages.

The Arabic language is widely used. Billions of people throughout the globe reportedly read and write in the Arabic script. Having access to OCR technology that can read Arabic characters would be very beneficial to businesses. Although Arabic writing is cursive, therefore, AOCR implementation phase faces significant technological challenges, particularly in the segmentation phase [4].

There hasn't been much advancement achieved, even though numerous academics are looking at potential remedies. A few academics have concentrated on deep learning for training the OCR framework, while others have concentrated on edge detection, flattening, and predictive analysis [5].

Since Arabic is transcribed from the right side to the left side in cursive handwriting, recognizing must take place in this direction. The Arabic letters consist of 28 different characters. Depending on where it is located throughout the words and subword, every character may take on between 2 and 4 alternative shapes. This results in a total of 100 distinct categories. Isolated letters are shown in their most fundamental form in Figure 1.

Figure 1: Basic Isolated Arabic characters

Each Arabic word might be separated into two, three, or even four different subwords. The "Zigzag(s)", "Dot(s)", "Kashida", and "Madda" that accompany various characters might appear below or above, or within the character itself. The general outline of many different characters is the same. The distinction is solely in the placement or quantity among those tertiary dots and strokes. For instance, the dots below and above the first and second letters of the word make them different from one another.

In Arabic, words often run together horizontally, and letters often pile up one on top of another. Both character and word segmentation are negatively impacted by this. There should be no uncertainty at this point why segmentation seems to be a necessary precondition to creating an AOCR framework.

Ligatures have an important role in writing Arabic. Characters like "Madda" and "Diacritic objects" are examples of ligatures since they share a horizontal region, resulting in overlapped vertical connections or separations. They make it harder to distinguish between the text lines and split them up into individual words. The letters of the Arabic alphabet often share similar exterior objects, such as the "Dots" or the "Hamza".

Many distinct fonts exist, and each one allows for a wide range of character lengths, line thicknesses, stroking qualities, and even horizontal letters shared due to slants. Because of the context of the words, the letter's form changes as you go around the word. Whether the character has been placed at the start, the ending, in the center, or by itself, it is rendered uniquely.

As shown in Figure 2, 15 characters of the Arabic language are distinguished by 1, 2, or 3 "Dots" positioned below or above the character block.

Figure 2: Various representations of Dots

The "Hamza" appears in 4 distinct places on the 4 characters shown in Figure 3.

Figure 3: Various representations of Hamza

Both dissection and recognizing methods may be used to carry out character segmentation. Dissection is the process of breaking down a larger image into smaller, more manageable parts by identifying and separating commonalities between them. Examining the image by separating it down into its parts is a key part of this process. Each segment of an image is processed in the same way as a letter for the sake of recognition. It's important to note how character categorization occurs after the fact.

Dissection methods employed by OCR frameworks include projecting evaluation, linked element computation, white spacing and pitching termination, and others. These methods work well with scripts that use blank lines amongst each character. Dissecting cursive writings would necessitate a highly sophisticated and tailored approach to assessment.

While efforts are being made to improve segmentation performance, there isn't any assurance that this will be successful. A "Mobility Window" of varying diameters is used as the foundation of recognizing character segmentation, with the preliminary segmentation being verified or rejected by the categorization. Techniques that separate characters based on this approach generate characters as a consequence of character identification [6].

The most notable benefit of this method is that it avoids the need of dealing with complex character-separating issues. In theory, there is no requirement for a script-specific segmentation technique, and identification mistakes are often the result of problems even during the classifying process. This is why this method is being used by an increasing number of OCR frameworks for cursive writing.

The task of creating an innovative, strong autonomous AOCR for the Arabic script has yet to be accomplished. Numerous aspects of the Arabic script are mostly responsible for this. Some Arabic characters even have loops, and the majority of the alphabet relies on dots for differentiation. Any one of those dots, if misread, may bring off the entire representation of the letter and, by extension, the entire sentence.

In comparison to more conventional methods of categorization and identification, CNN provides allowed for fully automated end-to-end solutions for AOCR. The CNN clearly shows superior performance compared to the best methods currently available [7]. A well-defined sequencing of layers is necessary for the successful development of a reliable CNN model. Even more importantly, it calls for optimizing strategies and parameterizing to enhance performance [8].

Problem Statement: Regardless of the extensive study on Asian and Latin language character identification and also the beneficial outcomes gained in Latin texts, there is still a long way to go. Only a small number of scholarly articles and publications focused on cursive AOCR exist. Studies on AOCR are picking up steam, and researchers have even documented making some efforts. Much existing literature doesn't quite address all elements of AOCR (whether handwritten or transcribed), rather than focuses on certain features.

Paper Contribution: This work aims to introduce a strong ECNN modeling approach for AOCR. In several ways, this work's significance is vital. To begin, research initial investigations showed that the AOCR issue may be solved by CNN modeling using a very large number of stacking layers with a significant degree of generalization. In this work, we enhance the CNN classifier by using "Batch Normalisation" and "Dropout Regularization" parameters to retrieve the most relevant features from their context. The objective above is to avoid the overfitting of data to the framework and to improve generalization capability over preexisting frameworks. Thorough testing on a standard dataset shows that the model outperforms other AOCR methods in terms of classification accuracy.

Paper Organization: This research article is organized as follows: Section 2 discusses certain latest publications concerning the issue of AOCR; Section 3 describes the suggested methodologies briefly with its modules, together with crunchy information of previous techniques; Section 4 exhibits the outcomes and comparing procured both for proposed and existing methods that use various criteria, and Section 5 provides a summary and conclusion.

2. RELATED WORKS

In [9] the developers compiled a customized "Arabic Handwritten Character Dataset (AHCD)" with the participation of 60 users. There are a total of 16800 letters in AHCD. By employing

"2 CLs" in their CNN framework, developers were able to increase the accuracy of classification to 88%. Regularization and other forms of optimization also were applied to the CNN framework to boost its effectiveness. The accuracy of their assessments generally stands at 94.94%.

For identifying Arabic numerals and letters, the developers in [10] created the "VGG-Net" framework. This same framework includes "13 CLs", "2 MPLs", as well as "3 FCLs". Two techniques such as "Data augmentation" and "Dropout" have been employed to prevent the overfitting issue from occurring. Both the "AD Base for the Handwritten Arabic numerical categorization subject" and the "HAC DB for the Arabic Handwritten characters identification subject" databases were used throughout the model's training and evaluation processes. The system obtained an ADBase precision of 99.66% and a HACDB precision of 97.32%, correspondingly.

The "Alexnet," a widely employed CNN framework, was employed by the developers of [11]. This comprised of "5 CLs", "3 MPLs", as well as "3 FCLs". The "OIHACDB-40" and the "AHCD" datasets have been used for the research. They used "Dropout" and "Minibatch" to train the CNN, and with properly tuned model parameters, they were able to improve the precision of the CNN to 98% only for "OIHACDB-40" and 97% again for "AHCD".

The CNN developed by the developers of [12] can identify "Devanagari, Persian, Urdu, Western Arabic, and Eastern Arabic" among other written scripts. After "2 CLs" are deployed to its source image's "28*28" pixels, an "MPL" procedure is performed on each convolutional-layers. An aggregate multilingual dataset achieved a precision of 99.26%. Overall, the mean precision in every language comes close to 99%.

To facilitate the directed production of individual Arabic Handwritten letters, the developers of [13] suggested a "Conditional Deep-Convolutional Generative Adversarial-Network (CDCGAN)". The "AHCD" database has been utilized to train the CDCGAN. The accuracy difference among both authentic and artificially created Arabic Handwritten letters was depleted to 10%.

3. METHODOLOGY

The main components of the proposed AOCR method are highlighted in Figure 4's schematic illustration. In this case, we employ BaseLine Segmentation to differentiate the text region, using ECNN as a classification for classifying Arabic characters across all classes. Several steps make up the presented AOCR mechanism's technique. Several stages are quite standard throughout OCR procedures. In this research, we focus on the two-stage process of segmentation and classification.

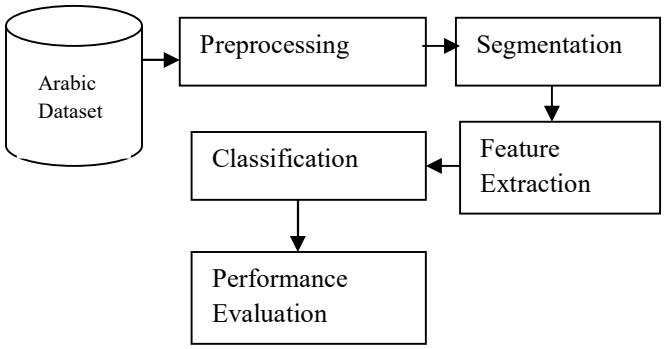

Figure 4: Proposed AOCR Methodology

3.1 PREPROCESSING

An accurate identification procedure might be affected by the source textual image quality. Improving the clarity of the original image file has been the responsibility of the preprocessing step. Its goal is to generate a flawless textual image that may be used effectively in subsequent stages. Normalizing and Centralization are two methods that may be used to complete the preprocessing operation. After getting the character image from the dataset as input. The image must be normalized and consolidated in an attempt to obtain the highest recognition accuracy. The image dimension is resized to a set dimension by normalizing. Most of the images in this category have been scaled down to 64 by 64 pixels and transformed into grayscale coloring maps. Characters appeared in numerous orientations in many of the images (left, right, bottom, and top). To begin, the character's and images' centroids are computed independently in an attempt to align all of the characters around the same place and compute the correct characteristics

for each Arabic-characters. Because the character is 64 by 64 dimensions in scale, the character's center point is 32 by 32 in this scenario. The character's centroid is therefore moved to the image's centroid to create a centered image. Finally, reformatting the image to applicable measurements to manage the dimension issue because a few of the characters seem to be smaller.

3.2 SEGMENTATION

When attempting to interpret Arabic characters, segmentation is an essential second stage. If the fundamental form of Arabic characters is not correctly segmented, a variant depiction of the character's element would be generated. The width of handwritten Arabic characters at a point of connection is always substantially narrower than that of the width of its preceding character. The proposed "Baseline Segmentation" method relies heavily on this characteristic.

The interconnections seen between Arabic characters are made along a middle line called the baseline. Equation (1) states that when the word is projected vertically using bi-level pixels.

$$v(j) = \sum_{i} w(i,j)$$

Eq→1

The connecting points would get sums that are below the "Average Value (AV)" according to Equation (2), whereby "w(i, j)" could be 0 or 1, whereas "i" represents rows and "j" represents columns.

$$AV = (1/Nc) \sum_{j=1}^{Nc} Xj$$

Eq→2

The proportion of black pixels within the "j^{th}" column is denoted by "X_j", wherein "N_c" is the total sum of columns.

Therefore, the boundary across characters would exist at the point where the total of the parts is significantly lower than AV. Figure 5 shows what happens if the vertical projection's resulting histogram doesn't at all satisfy the criterion of Equation (3) and hence fails to segment the character.

Using Equation (3), we recognize that the overall distance among consecutive peaks in Arabic characters never exceeds 1/3 of the character width.

$$|d_k| < d_l/3$$

Eq→3

In this case, "d_k" has been the distance between both the "k^{th}" and "k+1 peaks", and "d_l" is the whole character width. It is also expected that Equation (4) will hold after a subword or a word.

$$L_{k+1} > 1.5L_k$$

Eq→4

From that "L_k" denotes the histogram's "k^{th}" peak point. As a result of the interconnections of Arabic characters as well as their forms at the termination of a word, the above rule is applied.

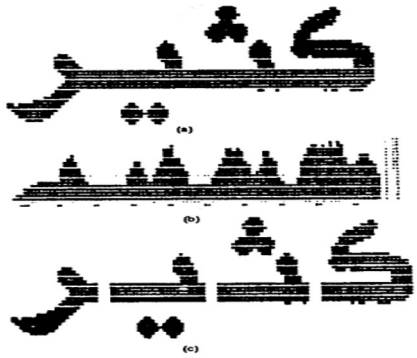

Figure 5: Arabic word segmented into even more characters: a. Word in Arabic, b. A histogram, and c. A segmented version of the word

Whenever characters are not yet segmented, such segmentation may be made by following the word's outside contour and thereafter measuring the distance across the two most distant spots where the contour meets a vertical line. This segmentation process is determined via a right-to-left horizontal scan of the confined contour through a movable window of width "w".

We determine the averaged vertical displacement all across the window termed as "h_{av}" for every one of the possible window positions. The accompanying criteria must be satisfied at the boundary point between two characters:

- **hav<T:** For this scenario, the mean vertical distance across the window must be smaller than a predetermined "Threshold (T)" in favor to classify the area as silent.
- As far as possible, recognized boundaries must simultaneously be on a very similar horizontal line, or the base line.
- When there's supposed to be a pause, no complimentary characters should appear below or above the baseline.

Whereas if the segmentation process produces a discarded character structure, it might be necessary to modify parameters "w" and "T" and do some backtracking.

3.3 FEATURE EXTRACTION

Following segmenting, the extraction of features stage's primary objective is to maximize detection performance with the fewest amount of features contained in a vector space. The objective of this process is to extract different various features from the character segmented image which have a significant level of resemblance between sampling of the respective classes and a significant level of variance between samples of different classes. In practice, APCA is frequently employed to find a lower-dimensional description of Arabic-Character datasets. There are 2 unique features of it. When processing the Arabic-character dataset that has been segmented, this generally maintains reducing the dimensions to a reasonable and correct range. First, it discretely reduces the total dimensionality by separating the number of distinguishable character features from partitioned Arabic-Character sources. The critical features have been maintained, allowing them to be utilized to determine whether the facts provided by the Arabic characters remained authentic. Employing the set of best Arabic-Character features allowed us to recover the covariance-matrix first from the matrix. From such covariance-matrix, the Eigen-values may be determined. Whole Arabic-Character datasets might be accurately represented by eigenvectors. The accessibility to data fluctuations is, however, fairly restricted, and just a few minor Eigen-values are much superior and bigger in relevance. Therefore, the preferable and higher variance solutions could easily be preserved in our previously proposed APCA by calculating the internal product of our Arabic-Character dataset well enough with corresponding Eigen-vectors for its corresponding Eigen-values.

3.4 AOCR CLASSIFICATION

3.4.1 EKNN

The Arabic Dataset validated the performance of this system, which incorporates an EKNN model for AOCR Categorization and also an EPSO for Selecting Features [14]. The "Inertia Parameters (w)", "Position Updating Method (PUM)", and "Fitness Function (FF)" all have an impact in determining how effectively PSO performs. It used scientific refinement and alterations to the PSO settings to achieve this. The weightage when after modifications could affect the characteristics of the EKNN classification model and drastically enhance the accuracy of classification due to the complete tuning of the processing parameters for PSO. Depending on FS input from EPSO, EKNN would be an improved classifier using a weighed KNN. EPSO utilizes LOOCV to determine the best possible weight for k and to assess EKNN's ability to classify data accurately. Inside the LOOCV, each data set represents its class. At each step of the process, one group would serve as a test sample, whereas the others were utilized for training. Despite compromising classification precision, the FS was able to significantly cut down on the amount of time it took for classifiers to calculate results and eliminate features that weren't ideal.

3.4.2 FKNN

Different classifiers are often used to compute a correlation between a set of objects based on those distance measures. The KNN approach was among the simplest forms of categorization without dimension minimization. When using Objects Membership Features, the selection method for the regular classifier treats all similarities equally. The "Fuzzy Logic (FL)" concept has been fused further with the "KNN" method to create the "Fuzzy-KNN (FKNN)" technique, which improved the categorization rate

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

of Arabic characters [15]. It's been demonstrated that the FKNN in addition to higher trust in the categorization availing use of the FL theory, as well as decreased inaccuracy in the categorization of Arabic Characters. The FKNN provides an object's class membership by providing a more useful matrix again for the object's membership. This method selects the most frequent representation of a given class based on the KNN. Decision-making could conduct ambiguous decisions by using FKNN to designate the sample's fuzzy memberships.

3.4.3 ECNN

a. CNN

CNNs are a kind of "Deep Neural Networks (DNNs)" that have achieved remarkable results in image analysis challenges. CNNs have many different uses, such as object recognition, imaging context segmentation, target tracking, and so on. A CNN's processing of its data collected follows a grid like architecture. One way to conceptualize an image's data input has been as a grid of dimensions "D*D*C", whereby "D*D" was an image's pixel quantity and "C" is the pixel's channel quantity "Grayscale=1 and RGB=3". Several layers, including "Convolutional Layers (CL)", "Pooling Layers (PL)", and "Fully Connected Layers (FCL)", receive this input grid and process it accordingly.

(i) Convolutional-layer

CNNs inherit the convolution function from the statistical convolution process, albeit their implementation of the term differs from the arithmetical and technical definitions. A minimum single NN layer undergoes convolutional transformations. Multiple filters, denoted by "K" make up a CL. The "K or kernels" is a common abbreviation. Every filter is a detector for a certain kind of feature, including endpoints, edges, or corners. The filter is often a squared grid with dimensions "N*N*R", in which "N" is the filter's width and height as well as "R" has been the channel count in an image. As every filter is pushed across the input grid, it multiplies every pixel by its associated value. Then, the products of each multiplication are combined to get a final tally. Features maps are the result of each convolution. When "K convolutional filters" are used, a collection of "K feature maps" of size "D-N+1" is produced. A numerical procedure known as an "Activation Function" is then applied to the feature maps. This will be performed so that classes that can be divided into linear subgroups may be identified. The "Sigmoid Function", the "Rectified Linear Unit (ReLU)" function, and the "Tanh function" are only a few examples of "Activation Functions" found in the research literature.

(ii) Pooling-layer

A PL often comes after the CL. To lower the image's resolution, pooling is usually conducted for every feature map individually. Different implementations of pooling exist, including the use of mean or max inside a size of the window "q*q".

(iii) Fully connected-layer

In addition to the CL as well as PL, there may be an arbitrary number of FCL. Every neuron inside an FCL is linked to every other neuron inside the layer above and below it. Finally, the network's output layers employ nonlinear groupings of information to produce estimations. Error reduction on unobserved instances is a primary focus in deep learning. In general, it's not ideal to have a model that does well on the known training data while poorly on the unknown test data. Regularization methods are a class of techniques designed to eliminate such mistakes. The "Weight Decay", and "Dropout" seem to be two common regularization techniques used in CNN models.

b. AOCR CLASSIFICATION BASED ON ECNN

Here, we detail the steps we took to enhance the Arabic dataset's AOCR functionality.

(i) Input-Layer (IL)

The IL is a pixel image "H*W*D", with "H" denoting the image's height, "W" denoting the image's width, and "D" denoting the image's depth. The D of an image's RGB is 3, whereas the D of an image's grayscale is 1. For our model to function, an image's grayscale with dimensions of "32*32*1" is required as input.

(ii) Hidden-layers (HL)

In a CNN, HL includes CL, PL, and FCL. In an attempt to process an image, the CL must first retrieve its features. It does this by extracting features like endpoints, edges, or corners and then imposing a suitable "Non-Linear" function,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

preventing the network from collapsing into a "Single-Linear" function.

In this framework, there are 3-CLs, and after each layer's "ReLU Activation Function," a "Batch Normalization" process is carried out. Several other CNN frameworks throughout the published research served as inspiration for our design. Due to the dimensions of our source image pixel's dimension "32*32", we have determined that the size of the filter of "3*3" is optimal. An obvious way to consider a filter is as a form of "Feature Detector".

The range of filters used by a given layer must approximately equal the range of features that is attempting to identify. For our inputs, we use a low-resolution grayscale image, and after some experimenting, we settled on a set of "64-filters" that seemed to do the task.

To maintain the network's representational efficacy as it becomes more complex, we augment the number of feature mappings following each PL. To prevent data degradation and image shrinking along the image's borders, we use padding in every CL.

Input to the first CL is a grayscale-image with a resolution of "32*32*1". For a CL of dimension "[((32+2*1) -3/1]+1=32", we employ kernels as "k=64 (3*3*1)", whereas zeropadding=1, as well as a stride=1. This yields a layer dimension of "32*32*64". When calculating "Activation Maps (i)", we utilize the following Equation (5):

$$Y_i^{(l)} = B_i^{(l)} + \sum_{j=1}^{k^{(l-1)}} W_i^{(l)} Y_j^{(l-1)}$$

Eq→5

Where,

l = "Current Layer",

$B_i^{(l)}$ = "Bias Matrix",

$k^{(l-1)}$ = "Previous layer's kernel quantity",

W = "Current layer's kernel matrix",

$Y^{(l-1)}$ = "Previous layer's output".

The ReLU function, specified in Equation (6), is our nonlinearity:

$$ReLU(x) = max(0, x)$$

Eq→6

After the first CL, a Max-PL with a window's dimension of "2*2*1" is applied, which produces a "16*16*64" layer. The next layer is a normalizing of the local reaction.

The subsequent CL has a layer dimension of "16*16*128", or features a map of "k=128", and uses a kernel dimension of "3*3*64", with the zero-padding=1, and a stride=1. A "ReLU Activation Function" typifies the nonlinearity. After this, we use a Max-PL with a window's dimension of "2*2*1", which produces a final array size of "8*8*128". The next layer is a normalization of the local reaction.

The final CL includes a features map with "k=256" and uses a kernel dimension of "3*3*128", with the zero-padding=1, and a stride=1, for a CL with dimension [((8+ 2*1)-3/1]+1=8, approximately with a layer of "8*8*256". A "ReLU activation function" is used for the nonlinearity. In its place comes a "Max-PL" with a window's dimension of "2*2*1", producing a total of "4*4*256".

After this comes a layer that adjusts for variations in local responses. Two further FCLs, of 1024 and 512 neurons, are then created from the tensor, after which the tensor gets flattened together into 4096-neuron FCL. To prevent overfitting, every FCL employs a dropout proportion of 80%, which is determined empirically.

(iii) Output layer (OL)

Upon this Arabic dataset, the OL is a 29-class "Softmax layer", with a class for every Hamza and the Arabic character. The dataset's OL is a "Softmax Layer" with 28-classes. Equation (7), wherein "N" represents the total of output classes, shows that the Softmax generates probability-like predictions for every character class.

$$softmax(x_i) = \frac{e^{x_i}}{\sum_{k=1}^{N} e^{x_k}} \quad \text{Eq}\rightarrow 7$$

c. Enhancement Process in Classifying Arabic Characters:

When using the "same" padding setting for every CL throughout this framework, the source image sizes would be preserved.

This approach is characterized in a preceding manner:

- To begin extracting convolutional features, we utilize the first CL layer, which has 32 feature maps, with each having a trainable kernel with dimension "3*3" pixels as well as a "ReLU Activation Function" of whole neurons.
- The "Grayscale", "RGB", and "RGBA" source raw images of dimensions "128*128" or "331*94" may all have features extracted from them, relying on the types of databases chosen inside this research.
- This CL is followed by a "Batch Normalization" layer, which serves as the second layer. It employed the variance and mean to restrict its outcome far from the saturation zone.
- Thirdly, we employed maximal subsampling, often known as PL, including a pool dimension of "3*3" to further reduce the feature maps' dimensions.
- Overfitting may be mitigated by using the "Regularization" or "Dropout" layer in the subsequent layer, this layer is set up to temporally and arbitrarily delete 10% of total neurons.
- A further 4 stacked blocks in succession reflected the arrangement of the subsequent repeating CLs.
- Following such convolutions, we normalized the resultant "2-D Feature Matrix" together into a single feature vector before moving on to the model's first classification layer.
- Then, it was sent to the initial FCL, which ultimately settled on a network size of "1024-neurons" activated by the ReLU function.
- After that, a layer of "Batch Normalization" with a "Dropout" of 0.5 was implemented.
- In the end, another FCL has been employed as the top layer of the structure.

The network was set up with the appropriate amount of neurons based on the labels in the desired Arabic database classes. The Softmax activation function has been employed for this last layer, and it provided results in the form of a probability distribution from 0 to 1 for every one of the target classes. Better classification performance from the CNN is possible through the use of "Batch Normalization" and the "Dropout" method. An overview of our framework is provided in Figure 6.

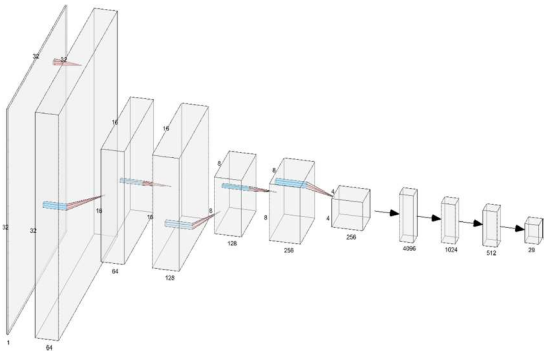

Figure 6: Proposed ECNN for AOCR

- A 2-CLs make up the ECNN used for AOCR. For the initial CL, we used a dimension of "28*28*80" since the source image's dimension inside the database was "32*32" pixels, and therefore no padding was used.
- Following this layer has always been a PL which shrinks the CL to a dimension of "14*14*80".
- Following a PL with dimension "5*5*64" comes the second CL, which is "10*10*64" in dimension.
- These are preceded by an FCL with dimension 1024, which would be preceded mostly by classifying OL.
- Each one of the CLs has used the ReLU function as their activation function, as well as the PL, would be a Max-PL with a "2*2" pixel window.

- After applying the Softmax function to the FCL, we get 28 distinct classes as expected from the research results.

4. RESULTS AND DISCUSSIONS

The developed ECNN technique has been subjected to a considerable range of testing. Below, we compare the efficiency of the newly proposed ECNN to that of the already existing EKNN and FKNN approaches. Moreover, the APTI Arabic datasets were used to analyze the Arabic Characters. "Matlab 2016" has been used to generate the required coding for the categorization process. In particular, it executed the experiments on a machine equipped with a "2.90-GHz processor with 16-cores", and "16-GB" of system memory. This is worth noting that output statistics often come in both analytical and descriptive forms. The best metrics including "Accuracy", "Precision", and "Recall", have been used within these research studies to assess a wide variety of parameter estimation in an attempt to validate these approaches.

(i) Accuracy Rate

Accuracy has been evaluated by calculating the confusion-matrix here between input data as well as the AOCR's outcome. The accuracy must be calculated using the formula below:

"Accuracy = (True-Negative + True-Positive) / (True-Negative + True-Positive + False-Negative + False-Positive)**"**

Table 1: Numerical Accuracy Comparision

Arabic Datasets	EKNN	FKNN	ECNN
Arabic Image-1	94.5	96.5	98.1
Arabic Image-2	95.5	97.5	99.2
Arabic Image-3	94.5	96.5	98.3
Arabic Image-4	93.5	95.5	97.9
Arabic Image-5	92.5	94.5	96.8

Most images throughout the databases undergo this process. The results are based on an analysis of AOCR data collected from the open-source dataset. Table 1 and Figure 7 show the maximum accuracy of the AOCR which was lower for the EKNN and FKNN existing approaches and higher for the proposed ECNN approach.

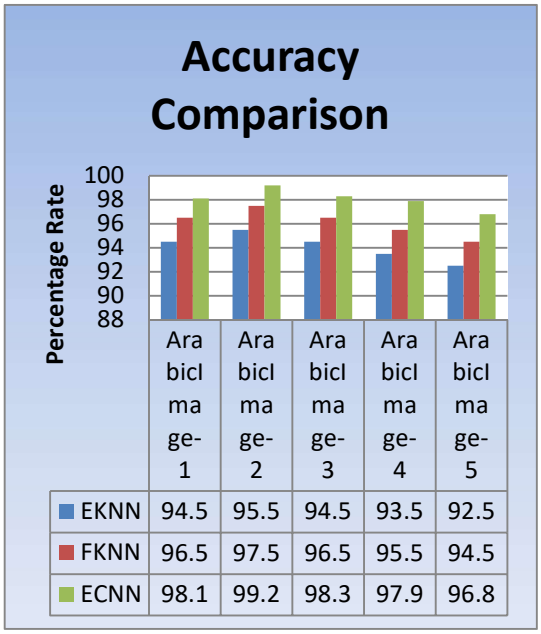

Figure 7: Graphical Accuracy Comparision

(ii) Precision Rate

In this context, "Precision" refers to the cumulative percentage of images properly categorized as corresponding to each type as determined by the AOCR for those images.

"**Precision** = (True-Positive) / (True-Positive + False-Negative)"

Table 2: Numerical Precision Comparision

Arabic Datasets	EKNN	FKNN	ECNN
Arabic Image-1	95	97	98.5
Arabic Image-2	96	98	99.5
Arabic Image-3	95	97	98.7
Arabic Image-4	94	96	97.9
Arabic Image-5	93	95	97.1

Most images throughout the databases undergo this process. The results are based on an

analysis of AOCR data collected from the open-source dataset. Table 2 and Figure 8 show the maximum precision rate of the AOCR which was lower for the EKNN and FKNN existing approaches and higher for the proposed ECNN approach.

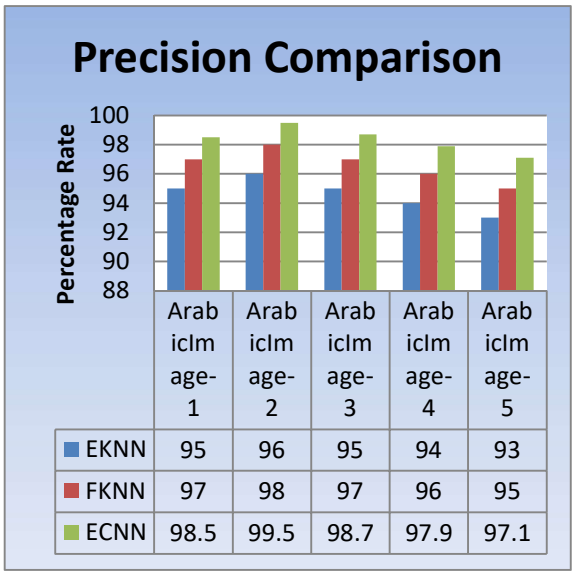

Figure 8: Graphical Precision Comparision

(iii) Recall

The percentage of images correctly attributed to a group out of the total number of images within this category inside the training set is known as Recall.

"**Recall** = (True-Positive) / (True-Positive + False-Positive)"

Table 3: Numerical Recall Comparision

Arabic Datasets	EKNN	FKNN	ECNN
Arabic Image-1	96.5	97.5	99.2
Arabic Image-2	97.5	98.5	99.8
Arabic Image-3	96.5	97.5	99.2
Arabic Image-4	95.5	96.5	98.7
Arabic Image-5	94.5	95.5	97.9

Most images throughout the databases undergo this process. The results are based on an analysis of AOCR data collected from the open-source dataset. Table 3 and Figure 9 show the maximum recall rate of the AOCR which was lower for the EKNN and FKNN existing approaches and higher for the proposed ECNN approach.

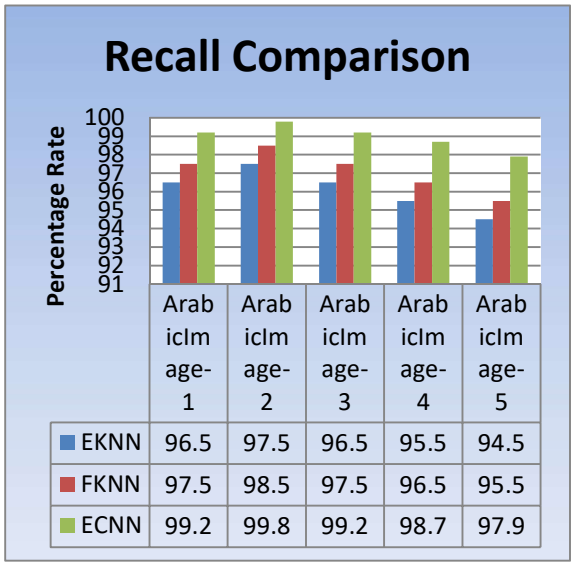

Figure 6: Graphical Recall Comparision

5. CONCLUSION

For this research, we put forward an ECNN framework specifically designed to identify Arabic characters. The framework was trained using an Arabic-language database. In this research, we developed a CNN-based intelligent method for AOCR. Applying "Regularization", we avoided the drawback of overfitting the model, and the "Dropout" process was implemented to further enhance the model's accuracy. The network was therefore trained independently upon every layer by using "Batch Normalization" to standardize the findings. The suggested methodology is a fusion of many "Stacked Deep ConvNets," as well as Dropout, pooling, FCL, and Batch Normalization. It was revealed that the model's functionality could be enhanced and the margin of error could be greatly reduced by using Regularization layers, Batch Normalization, and Dropout. The objectives of the approach are to have the quickest execution time, the simplest architecture, as well as the minimum number of identifying errors. While comparing with preexisting EKNN and FKNN systems, the suggested ECNN system has a higher rate of success in properly identifying a wide range of data

(99%) in a shortened amount of time. Multiple researchers show a great deal of interest in the AOCR. So, the suggested model may be used in coming future for additional identification challenges, including recognizing scribbled Arabic linked symbols or narratives, Arabic numbers, and so on. In the future, we plan to intend the bioinspired algorithm for AOCR classification.

REFERENCES

[1]. Al-Helali BM, Mahmoud SA. 2017. Arabic online handwriting recognition (AOHR): a survey. ACM Computing Surveys 50:1-35 DOI 10.1145/3060620.

[2]. Almansari OA, Hashim NNWN. 2019. Recognition of isolated handwritten Arabic characters. In: 7th international conference on mechatronics engineering, ICOM 2019.IEEE, 1-5.

[3]. Ali AAA, Suresha M. 2019. Arabic handwritten character recognition using machine learning approaches. In: Fifth international conference on image information processing (ICIIP). 187-192.

[4]. Balaha HM, Ali HA, Badawy M. 2021. Automatic recognition of handwritten Arabic characters: a comprehensive review. Neural Computing and Applications 33:3011-3034. DOI 10.1007/s00521-020-05137-6.

[5]. Boufenar C, Kerboua A, Batouche M. 2018. Investigation on deep learning for off-line handwritten Arabic character recognition. Cognitive Systems Research 50:180-195. DOI 10.1016/j.cogsys.2017.11.002.

[6]. Djaghbellou S, Akhtar Z, Bouziane A, Attia A. 2020. Arabic handwritten characters recognition via multi-scale hog features and multi-layer deep rule-based classification. Journal on Image and Video Processing 10:2195-2200.

[7]. Aljarrah MN, Zyout MM, Duwairi R. 2021. Arabic handwritten characters recognition using convolutional neural network. In: 12th international conference on information and communication systems, ICICS 2021. IEEE, 182-188.

[8]. Mustapha IB, Hasan S, Nabus H, Shamsuddin SM. 2022. Conditional deep convolutional generative adversarial networks for isolated handwritten Arabic character generation. Arabian Journal for Science and Engineering 47:1309-1320. DOI 10.1007/s13369-021-05796-0.

[9]. A. El-Sawy, M. Loey, and E. Hazem, "Arabic handwritten characters recognition using convolutional neural network," WSEAS Transactions on Computer Research, vol. 5, pp. 11–19, 2017.

[10]. M. A. Mudhsh and R. Almodfer, "Arabic handwritten alphanumeric character recognition using very deep neural network," MDPI, Information, vol. 8, 2017.

[11]. C. Boufenar, A. Kerboua, and M. Batouche, "Investigation on deep learning for off-line handwritten Arabic character recognition," Cognitive Systems Research, vol. 50, 2017.

[12]. G. Latif, J. Alghazo, L. Alzubaidi, M. M. Naseer, and Y. Alghazo, "Deep convolutional neural network for recognition of unified multi-language handwritten numerals," in Proceedings of the 2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR), pp. 90–95, London, UK, March 2018.

[13]. I. B. Mustapha, S. Hasan, H. Nabus, and S. M. Shamsuddin, "Conditional deep convolutional generative adversarial networks for isolated handwritten Arabic character generation," Arabian Journal for Science and Engineering, 2021.

[14]. V. M. Ashiq, Dr. E. J. Thomson Fredrik (2021). An OCR for Arabic Character Recognition with Ensemble Approach based Feature Selection for Enhanced KNN Classification. Design Engineering, 4728-4750. Retrieved from http://www.thedesignengineering.com/index.php/DE/article/view/5425.

[15]. V. M. Ashiq, Dr. E. J. Thomson Fredrik (2022). An OCR for Arabic character recognition with advanced principal component analysis based on feature extraction and fuzzy-KNN based classification. International Journal of Health Sciences, 6(S1), 12205–12224. https://doi.org/10.53730/ijhs.v6nS1.7918.