International Journal of Data and Network Science 8 (2024) 1501–1516

Contents lists available at GrowingScience

International Journal of Data and Network Science

homepage: www.GrowingScience.com/ijds

Integrated multi-layer perceptron neural network and novel feature extraction for handwritten Arabic recognition

Husam Ahmad Al Hamad[a*] and Mohammad Shehab[a*]

[a] *College of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan*

C H R O N I C L E

Article history:
Received: December 26, 2023
Received in revised format: January 29, 2024
Accepted: March 19, 2024
Available online: March 19, 2024

Keywords:
Arabic handwritten recognition
Block density and location feature
Pixel density
Feature extraction

A B S T R A C T

Arabic handwritten script recognition presents an energetic area of study. These types of recognitions face several obstacles, such as vast open databases, boundless diversity in individuals' penmanship, and freestyle writing. Thus, Arabic handwriting requires effective techniques to achieve better recognition results. On the other hand, Multilayer Perceptron (MLP) is one of the most common Artificial Neural Networks (ANNs) which deals with various problems efficiently. Therefore, this study introduces a new technique called Block Density and Location Feature (BDLF) with MLP, namely BDLF-MLP, which aims to extract novel features from letter images and estimate the letter's pixel density and its location for each equal-sized block in the image. In other words, BDLF-MLP can deal with various styles of Arabic handwritten, such as overlapping letters. The BDLF-MLP starts with the Block Feature Extraction (BFE) of the image by dividing the image into sixteen parts. After that, it calculates the density and location of each block (i.e., BDLF) by finding the sum of all values inside blocks. Finally, it determines the position of the greatest pixel density to obtain better recognition accuracy. The dataset containing 720 images is used to evaluate the efficiency of the proposed technique. Also, 1440 letters are used for training and testing divided evenly between them. The experiment results illustrate that BDLF-MLP outperformed the other algorithms in the literature with an accuracy of 97.26 %.

1. Introduction

The Arabic language is one of the oldest languages spoken by 25 countries with a population exceeding 423 million people. The number of letters in the Arabic language is 28 and each letter has more than one form according to its location in the word. Arabic language contains many punctuation marks, the meaning of the word changes according to the type of these marks, there are four main punctuation marks in Arabic: the "dammah", the "fatha", the "kasra", and the "sukoon". Arabic words are written in different forms and styles, some people write words in which the letters are staggered vertically or horizontally. All these features make the Arabic language different from other languages and give it a unique feature that requires different methods and techniques to distinguish its words and letters, especially handwritten words such as manuscripts and others (Guellil et al., 2021).

There are two types of handwritten letter recognition methods: online and offline, while the offline method uses scanned images of the letters, the online technique converts the strokes of the pen's tip into a list of coordinates (Al Hamad, 2012). The individual differences in writing styles make handwritten letter recognition difficult, even the same person's handwriting style can change. For offline handwritten letter recognition, traditional machine-learning techniques have been applied for a while. Image Pre-processing, word segmenting, feature extraction, and classification would be a typical machine-learning approach to handwritten letter recognition. A set of letters is first used to train offline handwritten letters. In case a new letter image is provided as input, it should be precisely recognizable by the technique. The recognition of handwritten documents

* Corresponding author.
E-mail address hhamad@aau.edu.jo (H. A. A. Hamad) moh.shehab12@gmail.com (M.Shehab)

ISSN 2561-8156 (Online) - ISSN 2561-8148 (Print)

doi: 10.5267/j.ijdns.2024.3.015

and forms, bank check reading, legacy document digitization, legal document digitization, and mail sorting in post offices are just a few examples of the many uses for handwritten letter recognition (Al Hamad, 2022). Digitization requires a difficult but necessary effort to convert information into a form capable of processing letter and word recognition quickly and accurately. When it comes to examining handwritten words and manuscripts that are often not legible and intelligible, gathering content and accessing texts becomes more difficult. The process of digitizing a handwritten document contains many steps including converting grayscale or color image to a binary matrix image, extracting the foreground matrix image from the background matrix image, removing the noise, separating the lines into individual words, segmenting the letters in each word, and recognizing the letters that are found individually in the word. There are many researchers employing various types of Artificial neural networks (ANN) such as convolutional neural networks (CNN), Multilayer Perceptron (MLP), Recurrent neural networks (RNN), more types are shown in Fig. 1. These types are utilized for recognizing and extracting the features, accurately. Each type has a specific mechanism, strengths, and weaknesses. Moreover, shortcomings of the applications of AI for abnormal detection, such as class imbalance and insufficient labelled fault samples.

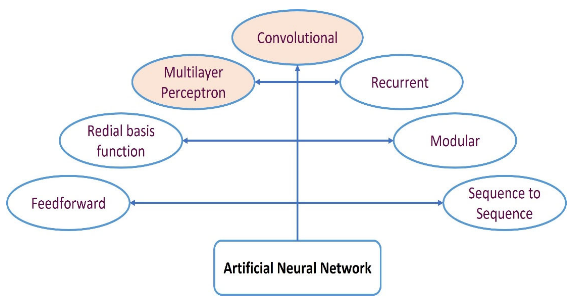

Fig. 1. Classifications of the Artificial Neural Network

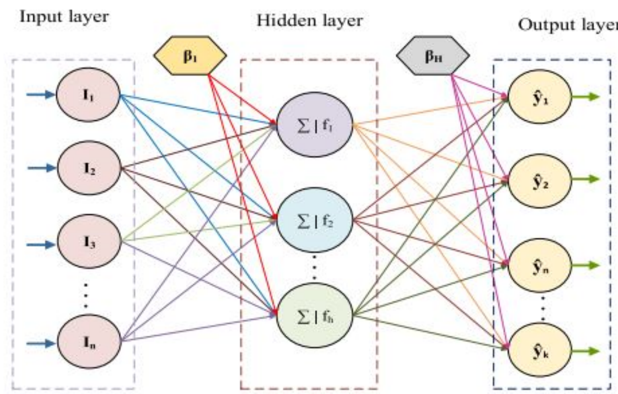

Fig. 2. MLP Structure (Ehteram et al., 2020)

The MLP neural network is one of the most widely used concepts in deep learning and machine learning. It is used in many applications, such as image classification and recognition, tabular datasets, prediction problems, and nonlinear problem processing. MLP is a multi-layered neural network in which all nodes of the network are interconnected with other layers, and all of its layers are connected. The network has many nodes in its layer, which increases the accuracy of the calculations because there are many parameters. The MLP demonstrates the ability to acquire broad internal representations for intricate nonlinear relationships (Xiang & Zhong, 2016; Altay et al., 2023). Comprising multiple layers, it includes an input layer, an output layer, and intermediate layers known as hidden layers, Fig. 2 illustrates the fundamental layout of an MLP. Also, it uses the back-propagation algorithm to increase the training g precision and reduce the error in the expected output and the actual output. The fundamental building block of an artificial neural network is the perceptron that determines the neuron in a neural network. It uses node values, activation functions, inputs, and node weights to calculate output in a supervised learning algorithm. Every node can reach the network only in the forward direction; does each node transmit its value to the one after it. MLPs can easily work with non-linear problems, and they can handle complex problems while dealing with large data sets and with the fitness problem of neural networks. Backpropagation has a higher accuracy rate and helps reduce prediction errors. After training the model, the multi-layer Perceptron neural network quickly predicts the output. A model will only perform well when it is trained perfectly. Due to the narrow connections of this model, the number of parameters and node redundancy increase (Al Hamad, 2013a; Moreno-Barea et al., 2020).

MLP is a flexible network that can be used to learn the layout from input through to output. This flexibility allows the network to be normalized on different types of data. For example, image pixel features in the image can be reduced to small features entered into the MLP as a vector. Additionally, document scripts can be converted to row data and put into the MLP as a vector. On other hand, CNN includes convolutional layers that are responsible for identifying features in a given image, such as curves, edges, and color. Classification predictions are inferred by aggregating connected network layers between the high-level features that generate them. The network works well with data that has a spatial relationship, as it is suitable for dealing with prediction problems that include image features as input. CNN has the advantage of being able to create an internal feature representation of a two-dimensional image, and this allows positioning and scaling to be known in different structures in the data. Numerous researchers have employed CNN in various fields, attributing their significant impact on image classification to various advantages. These include the capability to handle intricate images, recognize patterns in both clear and distorted images, cope with noise, and undergo training on one task for application to another. Furthermore, CNNs excel in automatically extracting features from images, eliminating the need for manual feature engineering, which is occasionally viewed as a limitation in specific cases. However, CNN has some weaknesses, such as a lack of spatial information, where it is information about the relative relationships between features (image entities). The maximum pooling layer was wrong as it was used to select the most active neurons thus losing most of the important information about the spatial locations of the entities. These shortcomings led to the creation of a capsule neural network as will be seen next. Another drawback of CNN is that it performs complex calculations, requires more time when training the network, which focuses on the presence of the object rather than its localization, and the difficulty in dealing with nested objects and segmentation. The reason for these drawbacks is the architecture used in CNNs, which involves learning image features through Feature Maps. This can lead to persistent loss of information position (Al Hamad, 2013b; Zhang et al., 2022). Furthermore, CNN is prone to overfitting, which means it can memorize the noise and details of the training data and fail to generalize to new and different data. To prevent overfitting, various regularization techniques must be applied, such as leakage, batch normalization, and data augmentation, which can increase the computational complexity and cost of the network. Another drawback of CNNs is that they are often considered black boxes, which means that they are difficult to interpret. This can pose challenges for debugging, validating, and trusting network decisions, especially in sensitive and critical areas such as healthcare, security, and law. Finally, CNN has a limited capacity to extrapolate novel situations. As a result, it might not perform well on images that are highly dissimilar from the training dataset. This may present a challenge in applications where CNN must handle a variety and large images. Table 1 shows a summary of the comparison between MLP and CNN.

Table 1
The main differences between networks MLP and CNN

Comparison	CNN	MLP
Data type	Image Data	Structural data
Parameters	Less parameters	Depends on data
Complexity	Complex	Simple
Layers	Convolution and Pooling	Neurons
Data volume	Big	Small and big
Weights	Respective weights	Weight matrices
Feature extraction	Automatic scalars	Input vector

This research focuses on classifications and deep learning to recognize offline handwritten Arabic letters. As mentioned in Table 1, it can be noticed that the mechanisms of MPL training and testing on the extracted features focus on the pixel density and location of each letter (i.e., it is characterized by the ability to train on pre-defined features). In contrast, CNN has prominent advantages in feature learning, can learn and extract features from data automatically. Therefore, the main contributions of this work are shown below.

- Introduction of a novel technique, BDLF-MPL, which leverages Block Density and Location Feature (BDLF) to improve the efficiency of MLP.
- Utilization of BDLF for feature extraction, followed by training and testing of MLP using the extracted features.
- Evaluation of the performance of BDLF-MPL using the Arabic Center for Document Analysis and Recognition (ACDAR) database.
- Comparison of the performance of BDLF-MPL with CNN using the same dataset.

The remainder of the paper includes the threats to a validity are shown in Section2. Section 3 presents the relevant works. The main procedures of the proposed technique are shown in Section 4. Section 5 illustrates the experiments and analysis of the results. Finally, the conclusion and future directions are illustrated in Section 6.

2. Threat to a validity

Essential in the handwritten Arabic recognition process, feature extraction plays a pivotal role by identifying pertinent details from raw data to capture patterns suitable for classification. Nevertheless, there exist potential threats to the validity of feature extraction within this specific context. The images provided as input might have disturbances like smudges, creases, or variations in lighting. If the feature extraction step is responsive to these disturbances, it could incorporate extraneous details, resulting in a decrease in recognition accuracy. Therefore, it is essential to eliminate or improve the quality of the noise before undertaking feature extraction. Approaches like image normalization, denoising, and contrast adjustment can prove advantageous in this regard. In addition, insufficient or partial preparing information may result in imperfect include extraction. On the off chance that the preparing dataset needs comprehensive scope of different written by hand Arabic varieties, the extricated highlights might not viably apply to unused, inconspicuous information. It is significant to ensure a different and agent preparing dataset. Improving the dataset by counting varieties in composing styles, sizes, and introductions can upgrade the versatility of the highlight extraction handle (Ghadhban et al., 2020). Another danger to the strategies for extricating highlights that perform successfully with a restricted dataset may not essentially scale proficiently when connected to bigger datasets. The highlights extricated must illustrate both vigor and versatility to handle the differing and complex nature of real-world written by hand Arabic information. To evaluate the adaptability of the include extraction procedure, it is pivotal to assess its execution over datasets of diverse sizes. Consequently, relieving these dangers requires a mix of utilizing changed datasets, tough highlight extraction calculations, reasonable preprocessing strategies, and fastidious consideration to variables impacting the reliability of extricated highlights within the domain of handwritten Arabic recognition (Dar et al., 2017; Song et al., 2022).

3. Literature Review

A parcel of investigation has been conducted on manually written letter recognition utilizing neural systems. This area presents the related works that talk about the recognition of transcribed scripts utilizing a few strategies and datasets for recognition and highlight extraction, taken after by Table 2 to appear the outline. Kavitha et al. (2018) utilized CNN to recognize transcribed Tamil characters in offline mode. The authors chose the CNN since it varies altogether from conventional Written by hand Tamil Character Acknowledgment (HTCR) strategies, fundamentally in its capacity to naturally extricate highlights from the information. The authors utilized a separated manually written Tamil character dataset that was created by HP Labs India. At that point, prepared a CNN show from scratch on this dataset in offline mode. The comes about illustrated that CNN accomplished a preparing exactness of 95.16%, which could be a noteworthy enhancement compared to conventional strategies of HTCR. Moudgil et al. (2023) utilized the CapsNet procedure to recognize the Devanagari characters. The Devanagari script comprises of 33 essential characters, 3 conjuncts, and 12 modifiers, collectively shaping the Devanagari letter set. The dataset was fastidiously categorized into 399 unmistakable classes, encompassing essential characters, modifiers, and conjunct characters, to encourage their precise recognition. The exploration included running the proposed show with different character preparation proportions and altering the number of preparing ages to optimize recognition exactness. It appeared that the most elevated acknowledgement exactness accomplished was a noteworthy 94.6% for recognizing Devanagari characters utilizing the CapsNet method.

In (Durga & Deepu, 2021), the authors introduced a new approach to pattern classification, leveraging a CNN that autonomously generates feature representations. These features, derived from handwriting patterns, underwent classification using an array of deep learning classifiers. The study evaluated the effectiveness of this group classification technique across various handwritten documents, demonstrating that it attained impressive classification accuracy, reaching as high as 90% for different patterns tested for a large number of words. Prieto et al. (2023) evaluated two distinct machine learning methodologies aimed at extracting data from historical tables containing handwritten content in pre-printed documents. The authors' investigation involves a comprehensive assessment of each approach's performance at various stages of the extraction process across diverse datasets. Furthermore, they drove the evaluation to a practical extreme by incorporating documents with disparate table layouts and handwritten content authored by different individuals. The findings showed that a Multilayer Perceptron-based model outperformed its counterparts when applied to more homogeneous documents. In contrast, a Graph Neural Network-based model demonstrated superior generalization capabilities when confronted with heterogeneous corpora. Ahmed et al. (2021) utilized a series of deep stacked-convolutional layers to construct our Deep Convolutional Neural Network (DCNN) architecture. Through extensive evaluations, the proposed model exhibits outstanding classification accuracy compared to traditional approaches in Optical Arabic Handwriting Recognition (OAHR). The evaluations span various benchmark databases, including Modified Arabic handwritten digits database (MADBase) (Digits), Center for Microprocessor Applications for Training Education and Research database (CMATERDB) (Digits), Handwritten Arabic Characters database (HACDB) (Characters), and Sudan University of Science and Technology- Arabic Language Technology group (SUST-ALT) (Digits). Additionally, the authors conducted experiments on Arabic databases using transfer learning (TL)-based feature extraction, showcasing the superior performance of the proposed model over state-of-the-art pre-trained VGGNet-19 and MobileNet models. Finally, they assessed the generalization capabilities of the models using the MNIST English isolated Digits database, affirming the superior performance of the proposed DCNN model.

Ali and Mallaiah (2022) introduced a new approach based on Back Vector Machine (SVM) and CNN for dealing with both single-font and multi-font sort recognition. To address the issue of overfitting, the authors actualized the dropout technique, which made a difference keep up the proposed approach's execution. Also, they presented an inventive training run the show for profound neural systems based on mistake back-propagation investigation. This run of the show pointed to playing down classification mistakes inside a most extreme interim. Besides, the authors combined max-margin least classification mistake (M3CE) and cross-entropy procedures to improve the, by and large, come about. To assess the proposed approach's execution, Arabic Manually written database (AHDB), Arabic Transcribed Character Dataset (AHCD), and written by-hand Arabic characters database (HACDB) databases have been utilized. The results showed that the proposed approach favorably comes about in content recognition. Meddeb et al. (2021) coordinated profound learning and profound interaction demonstrating encoders for suggesting Arabic literary records. The point of this integration is to extricate idle highlights related to clients and things. The authors tackled the control of profound user-item intuition by stacking different layers utilizing the created representations as inputs. This method illustrated its viability in capturing both the overarching setting and word recurrence data. Broad experimentation conducted on a standard Arabic dataset reliably illustrated the prevalent execution of the proposed procedure compared to state-of-the-art strategies. In (Singh et al., 2021), the authors utilized a covered-up Markov show to perform manually written word recognition in conjunction with a profound learning architecture known as a CNN. The word recognition framework within the ponder included both stroke-based category classification and word-based category classification approaches. They broadly analyzed the results of recognizing manually written words. Moreover, the proposed framework illustrated its capability to upgrade the routine comes about of penmanship recognition for different conspicuous Indian scripts. The discoveries of the consider showcased a recognition precision surpassing 97%, which the analysts considered satisfactory, outperforming the comes about recorded in existing writing.

In (Pareek et al., 2020), the authors highlighted the improvement of an offline Manually written Character Acknowledgment (HCR) framework utilizing manufactured insights particularly custom-fitted for the Gujarati dialect. An essential angle of this investigation is the broad information collection exertion, including 10,000 pictures sourced from 250 people having a place to differing age bunches and proficient foundations. The technique of the work utilized a directed classifier approach combining CNN and MLP for the reason of recognizing transcribed Gujarati characters. The results of the CNN-based demonstration illustrated an amazing precision rate of 97.21%, while the MLP demonstration accomplished a humble 64.48% exactness. Shuvo et al. (2022) presented a modern concept called Handwritten-to-Printed Numerical Recognition (HNR) with the basic suspicion that transcribed numerals can be seen as unmistakable varieties of their printed partners. The authors utilized auto-encoders and convolutional auto-encoders to perform the superimposition errand, changing Transcribed Numerical Pictures (HNIs) into Printed Numerical Pictures (PNIs). For the classification of these PNIs, neural systems and CNN are utilized. The execution of the proposed HNR framework is assessed on benchmark datasets containing transcribed numerals in Bengali, Devanagari, and English scripts. The outcomes about showed extraordinary recognition exactness, with HNR superimposition accomplishing recognition rates of 99.68%, 99.73%, and 99.62% for Bengali, Devanagari, and English transcribed numerals, separately. Souibgui et al. (2022) presented a modern approach to penmanship recognition based on few-shot learning for each letter set image. The methodology identified all symbols within a text line image pertaining to a specific alphabet and the subsequent interpretation of the obtained similarity scores to produce the final sequence of transcribed symbols. Then, the new approach undergoes an initial pre-training phase on synthetic line images that are generated from an alphabet, which may differ from the target domain's alphabet. Following this, a second training step is employed to bridge the gap between the source and target data. Also, the authors proposed an alternative solution that circumvents the need for extensive human effort through an unsupervised progressive learning approach. The approach's performance was evaluated across various datasets, demonstrating that it can achieve competitive results while substantially reducing the human effort traditionally associated with the annotation process.

Table 2
Summary of literature review

Ref.	Year	Technique	Description	Language	Dataset	Accuracy
Kavitha et al. (2018)	2022	CNN	Trained a CNN model from scratch on dataset in offline mode	Tamil	Private	95.16%
Moudgil et al. (2023)	2023	CapsNet	Adjusting the number of training epochs to optimize recognition accuracy	Devanagari	Private	94.6%
Durga & Deepu (2021)	2021	CNN	Evaluated the effectiveness of group classification technique across various documents	English	MNIST	90%
Prieto et al. (2023)	2023	Multilayer Perceptrons	Evaluated two distinct machine learning methodologies aimed at extracting data from historical tables	English	HisClima	-
Ahmed et al. (2021)	2021	DCNN	Exhibits outstanding classification accuracy compared to traditional approaches in OAHR	Arabic	CMATERDB, HACDB, MAD-Base	99.72%
Ali & Mallaiah (2022)	2022	SVM-CNN	Minimize classification errors within a maximum interval	Arabic	AHDB, AHCD, HACDB	98.58%
Meddeb et al. (2021)	2021	Deep learning	demonstrated its effectiveness in capturing both the overarching context and word frequency information	Arabic	BRAD1.0	95.2%
Singh et al. (2021)	2021	Hidden Markov	Enhance the conventional results of handwriting recognition for various prominent Indian scripts	Indian	Benchmark	97%
Pareek et al. (2020)	2020	CNN and MLP	Employed a supervised classifier approach combining CNN and MLP for the purpose of recognizing handwritten Gujarati characters	Gujarati	Private	97.21%
Shuvo et al. (2022)	2022	PNI-CNN	Utilized auto-encoders and convolutional auto-encoders to perform the superimposition task	Bengali, Devanagari, and English	Benchmark	99.73%
Souibgui et al. (2022)	2022	few-shot learning	Employed few-shot learning bridge the gap between the source and target data	-	Codex Runicus, Borg	99%
Alfaro-Contreras et al., (2023)	2023	self-supervised learning	Training a feature extractor using a neural network on a collection of unlabeled documents	-	TKH, GRPOLY, Capitan corpus	95%
Peng et al. (2022)	2022	CNN	Using transcript annotations, eliminating the need for character-level segmentation	Chinese	Benchmarks	94.76%

In (Alfaro-Contreras et al., 2023), the authors presented a novel self-supervised learning method designed to address a specific task. The approach involved training a feature extractor using a neural network on a collection of unlabeled documents. Then, applied this feature extractor to perform recognition tasks, requiring only a limited number of reference samples for each task. The experiments encompassed various types of corpora, including music, text, and symbol documents. The results indicated that the proposed method achieved high accuracy rates, reaching up to 95% accuracy in few-shot settings. Peng et al. (2022) presented a new method for handwritten Chinese text recognition based on segmentation using the CNN. In other words, the authors introduced a novel weakly supervised learning approach. This allows us to train the network exclusively using transcript annotations, eliminating the need for character-level segmentation. The proposed method was evaluated on four common benchmark datasets. The results showed the superiority of the proposed method in both online and offline handwritten Chinese text recognition.

4. Methodology

This section illustrates the main procedures of BDLF-MPL for handwritten Arabic recognition. The BDLF-MPL divides the letter into sixteen parts (16 blocks) and extracts novel features from each block based on the density (DF) and location (LF) of the foreground pixels of each letter, which is then trained using classifiers and computes letter recognition accuracy. As a first stage, the technique pre-processes the image, then extracts the density and location features, and thereafter, trains the novel features using neural networks, more details are shown in Fig. 3.

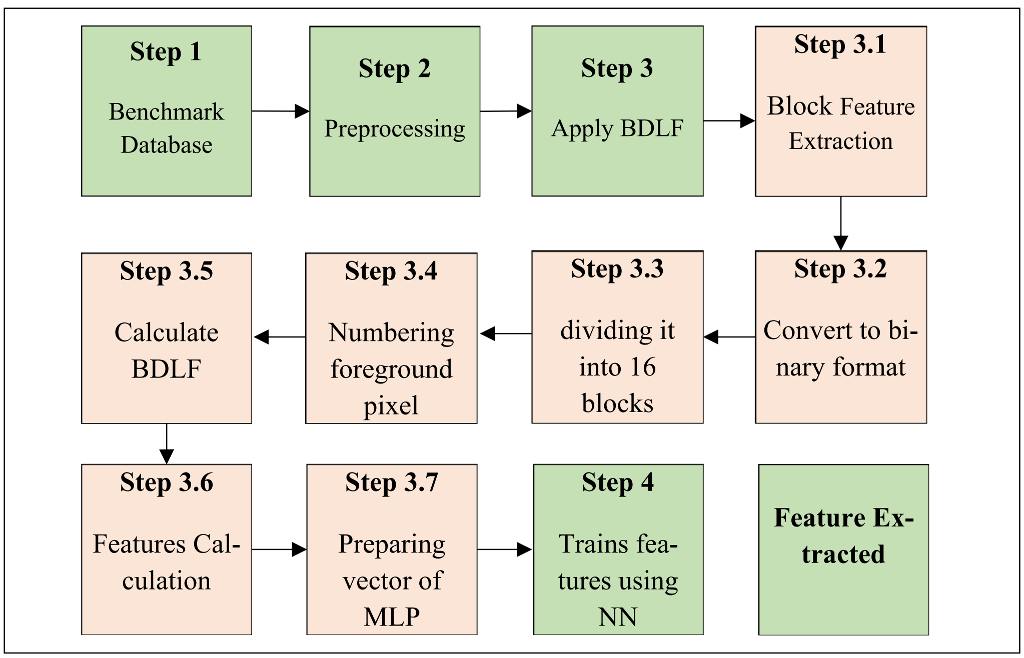

Fig. 3. Flowchart of the BDLF-MPL

The BDLF-MPL neural network was used for training and testing on the extracted features which focus on pixel density and location of each letter. CNN was also used to compare the results. It's worth mentioning that the MLP neural network is characterized by the ability to train on pre-defined features using BDLF, while the CNN extracts feature from word images automatically, which is more suitable for distinguishing complex images but has a disadvantage in that it is slow compared to the BDLF-MPL neural network. The time complexity of BDLF-MPL can be found in the following equation.

$$Time\ Complexity = n_{iter} \times (|P_T| + |P_V|) \times E\{t_{wait}\}$$

where n_{iter} refers to the number of iterations, P_T and P_V indicate the training pattern and validation pattern, respectively. $E\{t_{wait}\}$ refers to the mean delay value.

4.1 Benchmark Database

The Benchmark ACDAR database was used in the research. ACDAR includes a large number of Arabic paragraphs, lines, words, and letters. Handwriting paragraphs were written by 113 people and scanned in RGB. Writers are different in age, education, background, gender and country. The database contains 720 images (Al Hamad, 2015). Numerous earlier studies in literature have made use of the ACDAR database, which makes it easier to compare the findings with those of other studies and to assess how well the approach performed.

4.2 Pre-processing

Although several neural networks rely on colored images for training, such as the CNN network, the MLP network accepts features prepared in the form of vectors. The technique calculates features of pixel density and their location in the image. Removing the color levels helps better with image handling for some grids. Processing will be faster and less costly in terms of calculations and time. Although there are some pre-processing defects, such as the loss of some information from the original image or the occurrence of noise in the image. These defects will not have any effect on the accuracy of this technique, given that the extracted features depend on the density of blocks and their location, and this will be available if the image is color or binary. However, Better calculation speed and accuracy are what set dealing with binary pictures apart. For more accuracy, the technique through a normalization procedure removes small blemishes that are not considered part of the writing and removes noise elements for more accuracy in the results expected from the technique. The technique removes small objects that were not part of the letter and removes punctuation marks to improve letter recognition performance. The punctuation marks that are placed by the writer in different places pose obstacles to recognizing handwritten letters for many reasons, including affecting the pixel density of the blocks from one person to another and thus extracting inaccurate features that affect the accuracy of letter recognition.

4.3 Block Density and Location Features (BDLF)

The nature of the Arabic letters in terms of size and shape differ from each other. Some letters such as “Alif” have a small width with a large height, a letter such as “Sad” has a large width and a medium height, and another letter such as “Dal” has a small width and height. Some letters are of relatively long length, such as the letter “Yaa” or the letter "Sad" or the letter “Sheen”, which the recognition algorithms consider the letter as two parts. The proposed technique solves this problem by extracting the density and location feature (DLF), the following points show the main steps of BDLF.

1. Block Feature Extraction

To extract these features, the proposed technique divides the image of the letter into sixteen blocks, as shown in Fig. 4. From each block, features of density and location are extracted as block features. Through this division, the technique identifies the density of each block, as well as recognizes the features of the middle region of the letter, which is essential for the recognition of the letters. The technique also calculates the density of the letter in each corner and side of the image, which contains the left side, right side, upper and lower sides. Each block is considered also an additional feature to support the accuracy of letter recognition.

Fig. 4. Block Feature Extraction (BFE) of the image

2. Convert to binary format

After converting the letter into binary format, the technique assigns the foreground pixel (zero) see Fig. 5. The BDLF calculates the width and height of the letter to determine the number of blocks that should be created evenly. After that, the image is divided into sixteen parts to extract the advantages of density and location for it.

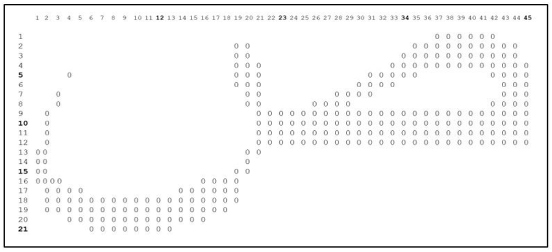

Fig. 5. Binary format of the Arabic handwritten letter “Sad”

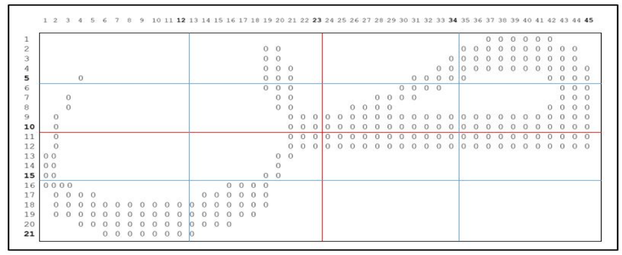

Fig. 6. The letter “Sad” after dividing it into 16 blocks

3. Dividing it into 16 blocks

After calculating the width and height of the letter, the technique determines the location of the sixteen blocks as shown in Fig. 6. Then it calculates the beginning and end of each block as integer numbers. If the width and height of MOD 2 are not equal to zero, this means there are extra columns or rows, these columns or rows are well placed on the left and bottom sides of the image, which are considered less important areas in Arabic letters as it is the end of the letter. In other words, the MOD 2 is used for special cases. For instance, if the width of the letter is 21 then there is a new extra row. Similar things happen in the height of the letter in case the height equals 5. Therefore, the increase will be added to the blocks at the end of the letter and not at the beginning.

4. Numbering foreground pixel

According to the block numbers specified in Fig. 6, the value of each foreground pixel (zero) in each block is replaced by the block number in the same order. For example, all zeros in block number 5 are replaced by the value 5, and so on for all other blocks. This means the value of each foreground pixel will be replaced by the corresponding block number. As a result, Fig. 7 shows the sixteen blocks after replacing their foreground pixel values with block numbers.

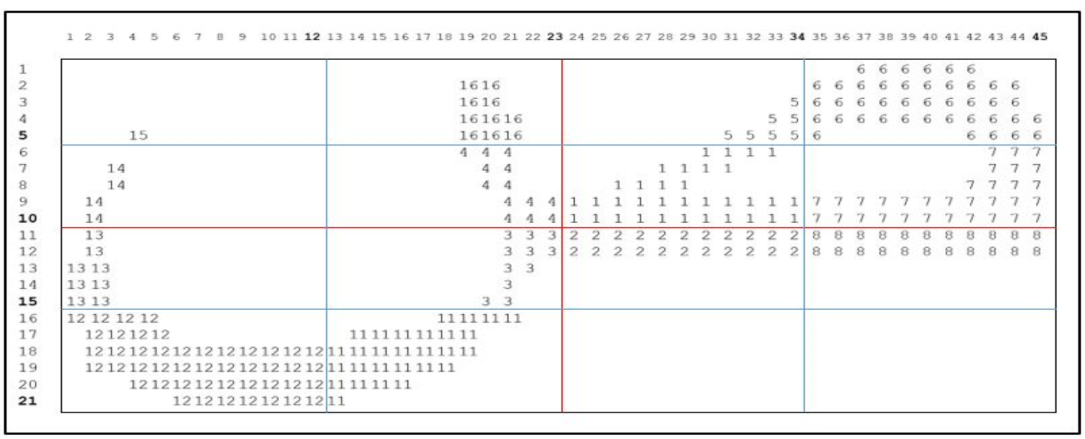

Fig. 7. The letter “Sad” after replacing the foreground pixels amount by block’s number

5. Calculates the density and location

The technique calculates the density and location of each block (BDLF) by finding the sum of all values inside blocks. To unify feature values, the feature value of each block is calculated between zero and one. For example, in block No. 6, the technique calculates the width and height of the block and collects all numbers inside it to find the density rate of font stroke according to its location. The feature value depends on the number of pixels that have values and the value of each pixel according to its location. The technique then calculates the density value between zero and one by dividing the resulting by 16. To clarify further, Fig. 8 shows an example of calculating the density of each of blocks 6, 7, 8 and 9. It appears in the figure that the value of the calculated feature of block 6 is 4.58 and after dividing it by 16, becomes 0.2846, and as shown in the figure, the final feature values of blocks 7, 8 and 9 are 0.2545, 0.2 and zero respectively. Values of zero indicate that there are no features in that block.

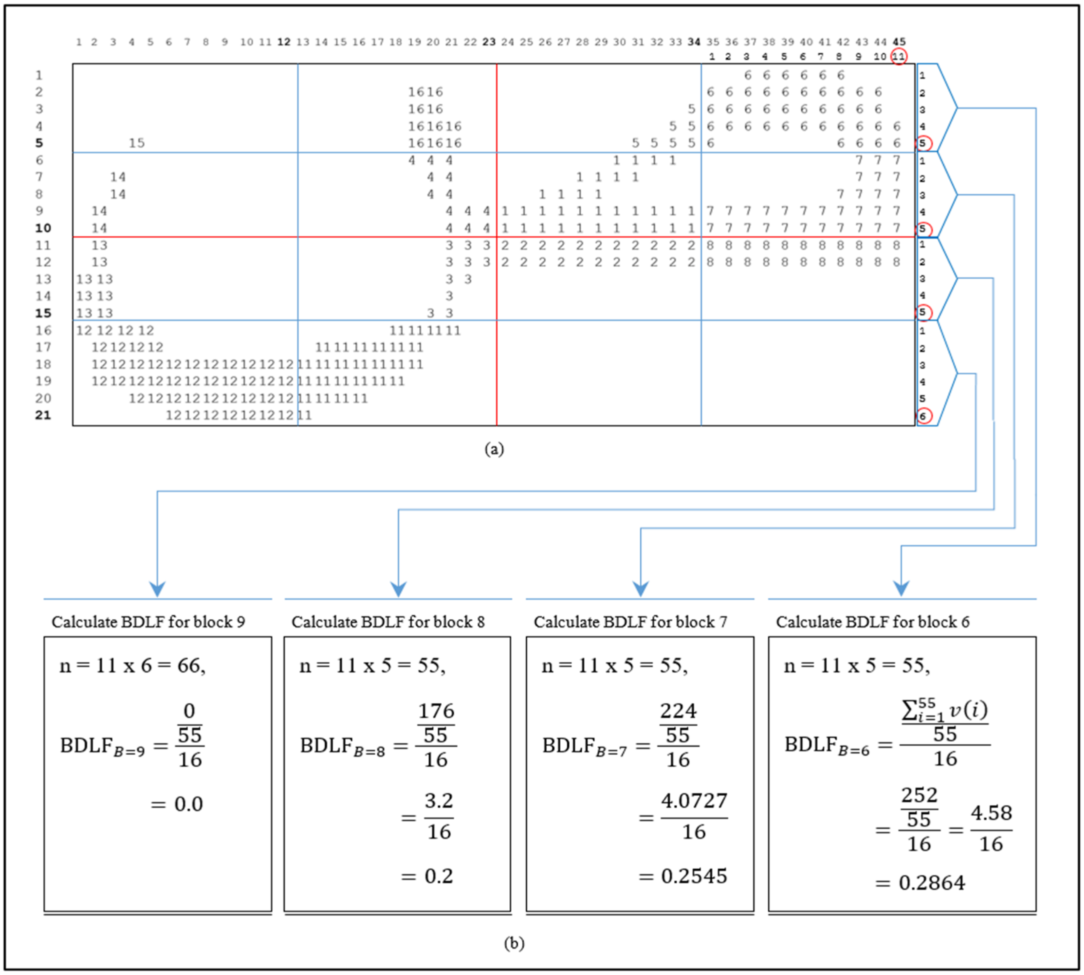

Fig. 8. Calculate BDLF for each block, a) Calculate the width and height of each block, b) An example of calculating features of blocks 6, 7, 8, and 9

Fig. 9 shows the result of calculating the density and location features of the letter "Sad". The computed block features will be part of the neural network vector that the MLP neural network will train. Other values of the vector will be computed within multiple combinations of the sixteen blocks, which include angles, and sides of the letter matrix as will be shown later.

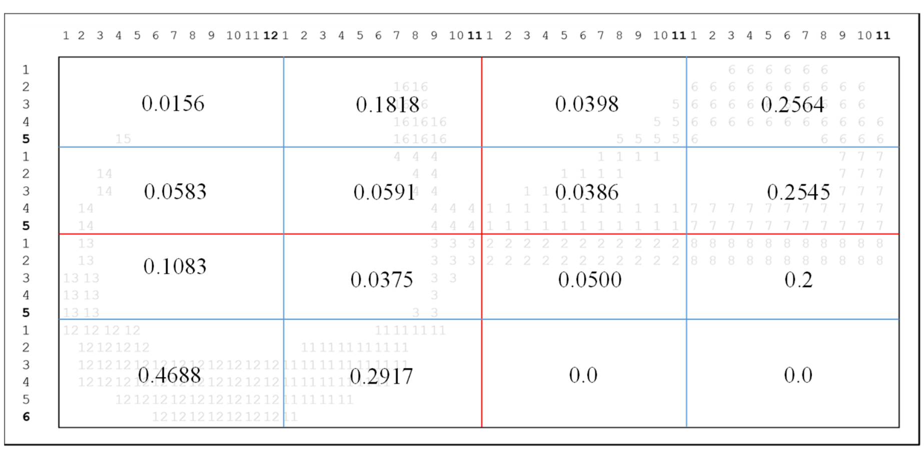

Fig. 9. The result of calculating BDLF for all 16 blocks of the letter "Sad"

6. *Features calculation*

The second part of the task that the technique focuses on is determining the position of the greatest pixel density in order to obtain better recognition accuracy. This method is suitable for Arabic letters that differ significantly from each other in terms of size, shape and dimensions. Some of these letters have density in the edges, middle, or any parts of the block. The proposed technique addresses all these differences. It also benefits them for achieving better accuracy. Additional features are extracted to achieve the highest accuracy for training the neural network. In addition to extracting the features of each of the sixteen blocks, the technique extracts and calculates the features of the center, and the four angles, which contain the right-top side, right-bottom side, left-top side, and left-bottom side. The technique also calculates the four sides, which contain top, bottom, right and left side features. This variety of features further determines the density and location of pixels in the image. Table 3 shows all the calculated features and will be part of the vector that the neural network will be trained on.

Table 3

Equations of features calculation of the corners and edges of the letter image

Feature	Location	Equation
Blocks from 1 to 16	**15 16 5 6** **14 4 1 7** **13 3 2 8** **12 11 10 9**	$\mathrm{BDLF}_{B=1\ to\ 16} = \frac{\sum_{i=1}^{n} v(i)}{n}/16$
Center blocks	15 16 5 6 14 **4 1** 7 13 **3 2** 8 12 11 10 9	$\mathrm{BDLF}_{C=1,2,3,4} = \frac{\sum_{C=1}^{4}\left[\frac{\sum_{i=1}^{n} v(i)}{n}/16\right]}{4}$
Right-Top corner blocks	15 16 **5 6** 14 4 **1 7** 13 3 2 8 12 11 10 9	$\mathrm{BDLF}_{\mathrm{RT}=1,5,6,7} = \frac{\sum_{RT=1}^{4}\left[\frac{\sum_{i=1}^{n} v(i)}{n}/16\right]}{4}$
Right-Bottom corner blocks	15 16 5 6 14 4 1 7 13 3 **2 8** 12 11 **10 9**	$\mathrm{BDLF}_{\mathrm{RB}=2,8,9,10} = \frac{\sum_{RB=1}^{4}\left[\frac{\sum_{i=1}^{n} v(i)}{n}/16\right]}{4}$
Left-Top corner blocks	**15 16** 5 6 **14 4** 1 7 13 3 2 8 12 11 10 9	$\mathrm{BDLF}_{\mathrm{LT}=4,14,15,16} = \frac{\sum_{LT=4}^{4}\left[\frac{\sum_{i=1}^{n} v(i)}{n}/16\right]}{4}$
Left-Bottom corner blocks	15 16 5 6 14 4 1 7 **13 3** 2 8 **12 11** 10 9	$\mathrm{BDLF}_{\mathrm{LB}=3,11,12,13} = \frac{\sum_{LB=3}^{4}\left[\frac{\sum_{i=1}^{n} v(i)}{n}/16\right]}{4}$
Top blocks	**15 16 5 6** **14 4 1 7** 13 3 2 8 12 11 10 9	$\mathrm{BDLF}_{\mathrm{T}=1,4,5,6,7,14,15,16} = \frac{\sum_{T=1}^{8}\left[\frac{\sum_{i=1}^{n} v(i)}{n}/16\right]}{8}$
Bottom blocks	15 16 5 6 14 4 1 7 **13 3 2 8** **12 11 10 9**	$\mathrm{BDLF}_{\mathrm{B}=2,3,8,9,10,11,12,13} = \frac{\sum_{B=2}^{8}\left[\frac{\sum_{i=1}^{n} v(i)}{n}/16\right]}{8}$
Right blocks	15 16 **5 6** 14 4 **1 7** 13 3 **2 8** 12 11 **10 9**	$\mathrm{BDLF}_{\mathrm{R}=1,4,5,6,7,14,15,16} = \frac{\sum_{R=1}^{8}\left[\frac{\sum_{i=1}^{n} v(i)}{n}/16\right]}{8}$
Left blocks	**15 16** 5 6 **14 4** 1 7 **13 3** 2 8 **12 11** 10 9	$\mathrm{BDLF}_{\mathrm{L}=3,4,11,12,13,14,15,16} = \frac{\sum_{L=3}^{8}\left[\frac{\sum_{i=1}^{n} v(i)}{n}/16\right]}{8}$

7. *Preparing vector of MLP*

After calculating the density and location features of blocks, center, angles, and edges, the technique prepares the vector of the neural network. The vector consists of 25 features as shown by the Eq.1 below. The vector size is relatively small compared with another research. This size supports faster computations and training of the neural network than suggested by another research. In addition, the accuracy of the features and their representation of letters will be reflected in the accuracy and speed of the algorithm.

$$\text{Total BDLF features} = SB\,[16] + CB\,[1] + CoB\,[4] + EB\,[4] = 25 \quad (1)$$

where *SB*, *CB*, *CoB*, and *EB* indicate the Single Blocks, Center Blocks, Corner Blocks, and Edge Blocks, respectively.

4.4 Neural Network

The same letters are trained and tested using two networks with an initial learning rate equal to 0.01. Feature extraction techniques BDLF is used to create appropriate input vectors for the MLP network. Feature extraction of the CNN network is calculated automatically after normalizing the inputs. For each set of experiments, the results are shown in tabular format. The results contain the number of epochs the network was trained for, training performance, training error, and the CPU time rate of the training. For the MLP network, the technique used a back-propagation algorithm with the fully connected neural network. The training is done using the Backpropagation algorithm with options for learning rate decrease, momentum back-propagation resilience, and gradient descent. The training of the neural network stops when the mean square error reaches zero or reaches a maximum number of epochs. To extract features in the CNN network, the hidden neurons in each layer learn non-linearly from the initial input. New features extracted from one layer become inputs for the next layer. The learned features become inputs to the regression function or classifier at the end of the network. The data was normalized by converting the image to binary format before training the network to improve the results and speed up the training of the network. The input images became binary format with a range (0, 1), with a mean equal to zero and a standard deviation equal to one, which leads to better results.

5. Experimental Results and Analysis

In this section, numerous experiments were conducted using MLP and CNN neural networks. The results were compared with each other and with the literature. All images were trained and tested using the features mentioned before. For the MLP neural network, the technique performed the experiments using a back-propagation neural network. The BDLF was extracted to generate novel input vectors. As for the CNN network, the features are extracted automatically according to the images of the input letter. Classifiers performance and recognition results are presented later for all experiments. The experiments deal with 720 individual letters that were extracted manually for the training features vector in the classifier and testing set 720 letter patterns. BDLF feature re-scaled from letter images with input vector contains 25 features passed to the MLP classifier, each BDLF feature contains 16 blocks, 1 center, 4 angles, and 4 edges areas. In total, 72 confidence values were among the outputs that were chosen for the neural network. Training time for each experiment was also calculated in seconds. It was considering computer speed with a Core i7 processor, 16 GB of RAM, and Windows 11. The results include several training epochs and training elapsed time. Also, the accuracy and error were calculated using the following equations.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP, TN, FP, and FN indicate the true positive, true negative, false positive, and false negative respectively. It can be noticed that the error percentage is calculated as a complement of the accuracy score. The parameters of MLP used in the experiments were set as follows; learning rate η=0.7, a satisfactory value for α is 0.9, u = 1.2, and d = 0.8. As expected, the accuracy of the MLP neural network was better than that of the CNN neural network. As shown in Table 4 shows the letter recognition results for MLP using the BDLF feature and CNN neural network, the highest score obtained for MLP is 97.22% and for CNN, it is 93.19% using 15 training epochs. CPU training time for each experiment is around 54 seconds for the MLP neural network, and for the CNN neural network around 78 seconds including input and output using 15 epochs.

Table 4
Performance of MLP and CNN neural networks

Experiment	Network	Epochs	Elapsed time (Second)	Error	Accuracy	Classification Rate Set
1	MLP	5	24.1562	6.11%	93.89%	676/720
2	MLP	10	38.0938	3.47%	96.53%	696/720
3	MLP	15	54.2812	2.78%	97.22%	700/720
4	CNN	5	36.9062	9.44%	90.56%	652/720
5	CNN	10	65.9375	7.36%	92.64%	668/720
6	CNN	15	78.5625	6.81%	93.19%	672/720

The results in Table 4 illustrate the relationship between the performance of neural networks and the number of epochs. It can be noticed that the neural network and feature type effect on accuracy and speed of processing. In addition, writing style also affects exerting features and then speed and accuracy of recognition. Consequently, the MLP network achieved better results than the CNN network in all experiments.

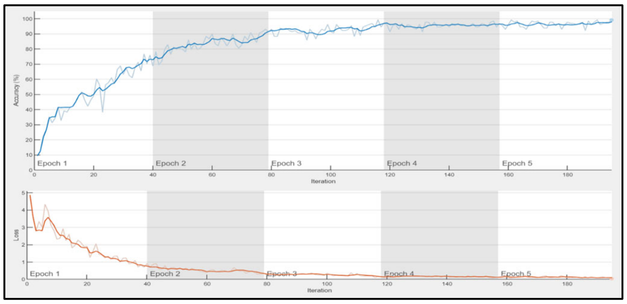

Fig. 10. Experiment results of MLP neural network at 5 epochs with accuracy 93.89%.

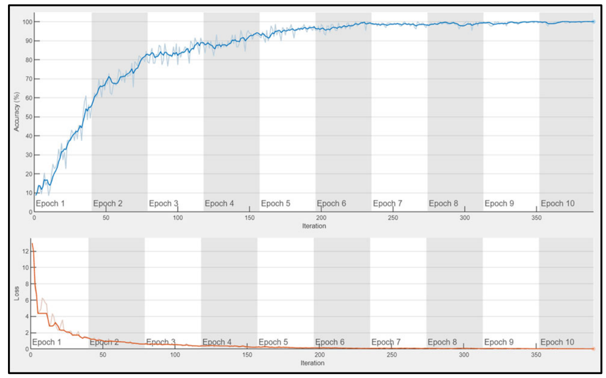

Fig. 11. Experiment results of MLP neural network at 10 epochs with accuracy 96.53%.

The experimental performance of the MLP neural network is shown in Fig. 10, Fig. 11, and Fig. 12. Fig. 10 illustrates the performance at 5 epochs, with an accuracy score of 93.89%. Fig. 11 illustrates the performance at 10 epochs, with an accuracy score of 96.53%. Fig. 12 illustrates the performance at 15 epochs, with an accuracy score of 97.22%. The experiment performance of the CNN neural network is shown in Fig. 13. It illustrates the performance at 5 epochs, with an accuracy score of 90.56%.

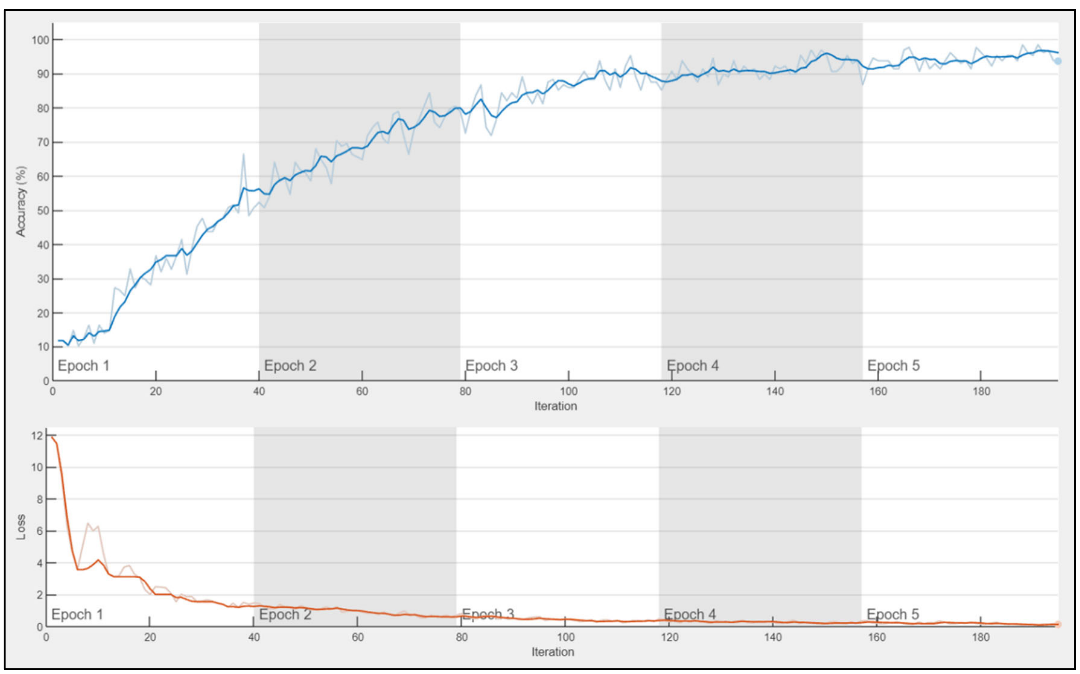

Fig. 12. Experiment results of CNN neural network at 5 epochs with a ccuracy 90.56%.

Fig. 14 illustrates the comparison of elapsed training time between MLP and CNN networks in seconds. As shown in the figure, the performance of the MLP was faster than the CNN neural network at relatively high rates. Using appropriate features vectors of MLP is considered a major reason for the network performance and speed. The training time of this network took approximately 24 seconds at epoch 5, and at epoch 10, it took approximately 38 seconds, and at epoch 15, it took approximately 82 seconds. While in the CNN network that automatically extracts the features, the network performance and speed were slower. At epoch 5 the network took approximately 37 seconds, and at epoch 10, it took 66 seconds, and at epoch 15, it took approximately 78 seconds. The increased percentages of the time spent in the CNN network compared with the MLP network were respectively 53%, 73% and 45%.

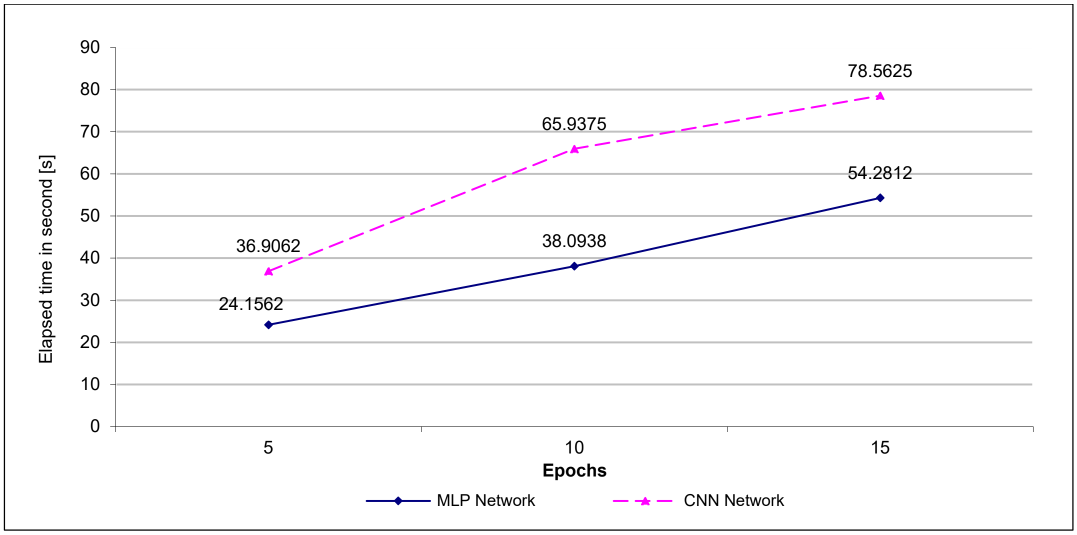

Fig. 13. Elapsed training time in seconds for MLP and CNN neural networks

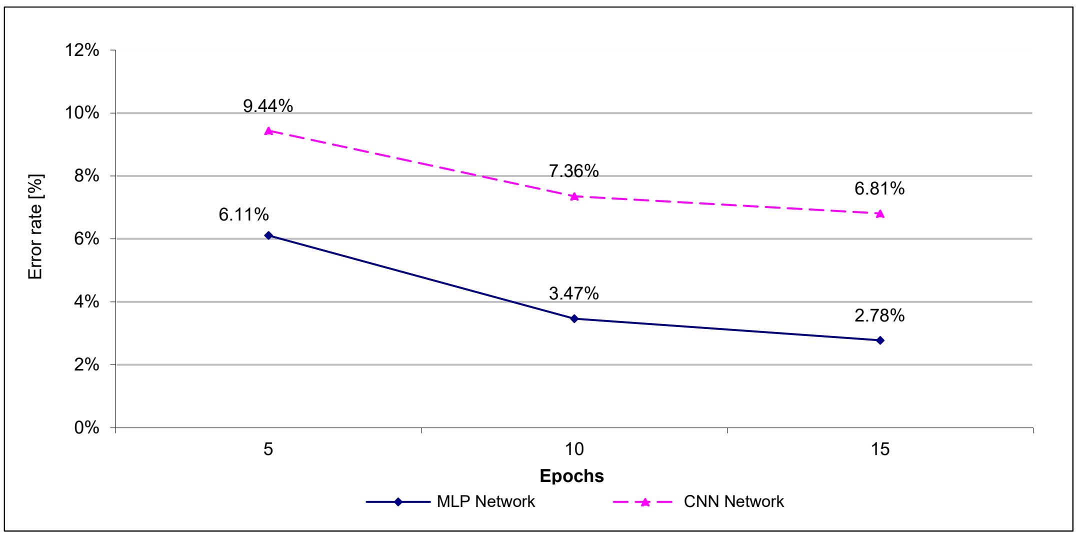

Fig. 14. Recognition error rate of MLP and CNN neural networks

Table 5 shows a comparison of the technique results with other techniques and algorithms in the literature. It is clear from the comparison, that the MLP neural network shows superior results compared to other similar research in literature that uses neural networks for handwritten letter recognition.

Table 5
Comparison results with other research in the literature

References	Year	Technique	Accuracy	Dataset
Nahar et al. (2023)	2023	Hybrid Model VGG16 + NN	88%	900 images
Balaha et al. (2021)	2021	CNN	90.70%	800 images
Gupta & Bag (2020)	2021	CNN (2 Layer)	96.23%	750 images
Rabi et al. (2018)	2018	Hidden Markov	87.93%	500 images
Ghanim et al. (2020)	2020	Matching & Rank-ing	91.60%	859 images
Maalej & Kherallah (2022)	2022	MDLSTM	92.59%	500 images
Haq et al. (2023)	2023	CapsNet	88.96%	425 images
Ali & Mallaiah (2022)	2022	CNN-SVM	86.47%	500 images
This research			97.26% MLP 93.19% CNN	720 images

6. Conclusion and future work

The proposed technique achieved an excellent performance in recognizing the Arabic letters. This study provides a brief explanation of the most widely used and well-known neural networks for recognizing images and handwritten documents, which are MLB and CNN. A new was introduced called Block Density and Location Feature (BDLF) with Multilayer Perceptron (BDLF-MLP) which aided in extracting novel features from letter images and estimating the letter's pixel density and its location for each equal-sized block in the image. Experiments showed promised results for the recognition of Arabic letters. The accuracy of overall letter recognition using the MLP network reached 97.22% compared with the CNN network which reached 92.64% at 15 training epochs. According to the criteria outlined in the works of literature research, the effectiveness of the entire letter recognition technique was assessed. The recognition accuracy of the technique turned out to be superior in comparison to other research. The technology also achieved good processing speed, as the use of new features reduced the training time of the neural network, which was 54.2812 seconds for the MLP network compared to 65.9375 for CNN neural networks.

Although the BDLF-MLP proved its efficiency compared with CNN. The authors plan to enhance the performance of CNN using a capsule network (CapsNet) in future work. It's worth mentioning that CapsNet can encode the characteristics and spatial information of features in an image to achieve equivariance by using a set of neurons as a capsule or vector to substitute the neuron in the classic neural network. In addition, making advanced preprocessing for CNN and More customization for CNN parameters to Apply the technique to complete sentences.

References

Ahmed, R., Gogate, M., Tahir, A., Dashtipour, K., Al-Tamimi, B., Hawalah, A., ... & Hussain, A. (2021). Novel deep convolutional neural network-based contextual recognition of Arabic handwritten scripts. *Entropy, 23(3)*, 340.

Alfaro-Contreras, M., Ríos-Vila, A., Valero-Mas, J. J., & Calvo-Zaragoza, J. (2023). Few-shot symbol classification via self-supervised learning and nearest neighbor. *Pattern Recognition Letters, 167*, 1-8.

Al Hamad, H. A. (2012). Over-segmentation of handwriting Arabic scripts using an efficient heuristic technique. In *2012 International Conference on Wavelet Analysis and Pattern Recognition* (pp. 180-185). IEEE.

Al Hamad, H. A., Abualigah, L., Shehab, M., Al-Shqeerat, K. H., & Otair, M. (2022). Improved linear density technique for segmentation in Arabic handwritten text recognition. *Multimedia Tools and Applications, 81(20)*, 28531-28558.

Al Hamad, H. A. (2013a). Use an efficient neural network to improve the Arabic handwriting recognition. Signal and Image Processing Applications (ICSIPA). In *2013 IEEE International Conference on. IEEE*, 269–274.

Al Hamad, H. A. (2013b). Neural-based segmentation technique for Arabic handwriting scripts. in *Proc. 21st International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision in co-operation with EUROGRAPHICS Association*, West Bohemia, Czech Republic, pp. 9-14.

Al Hamad, H. A. (2015). Skew detection/correction and local minima/maxima techniques for extracting a new Arabic benchmark database. *International Journal of Advanced Computer Science and Applications (IJACSA), 6(9)*, 1-10.

Ali, A. A., & Mallaiah, S. (2022). Intelligent handwritten recognition using hybrid CNN architectures based-SVM classifier with dropout. *Journal of King Saud University-Computer and Information Sciences, 34(6)*, 3294-3300.

Altay, O., & Varol Altay, E. (2023). A novel hybrid multilayer perceptron neural network with improved grey wolf optimizer. *Neural Computing and Applications, 35(1)*, 529-556.

Balaha, H. M., Ali, H. A., Saraya, M., & Badawy, M. (2021). A new Arabic handwritten character recognition deep learning system (AHCR-DLS). *Neural Computing and Applications, 33*, 6325-6367.

Dar, K. S., Shafat, A. B., & Hassan, M. U. (2017, June). An efficient stop word elimination algorithm for Urdu language. In *2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)* (pp. 911-914). IEEE.

Durga, L., & Deepu, R. (2021). Ensemble deep learning to classify specific types of t and i patterns in graphology. *Global Transitions Proceedings, 2(2)*, 287-293.

Ehteram, M., Ahmed, A. N., Ling, L., Fai, C. M., Latif, S. D., Afan, H. A., ... & El-Shafie, A. (2020). Pipeline scour rates prediction-based model utilizing a multilayer perceptron-colliding body algorithm. *Water, 12(3)*, 902.

Ghadhban, H. Q., Othman, M., Samsudin, N. A., Ismail, M. N. B., & Hammoodi, M. R. (2020). Survey of offline Arabic handwriting word recognition. In *Recent Advances on Soft Computing and Data Mining: Proceedings of the Fourth International Conference on Soft Computing and Data Mining (SCDM 2020)*, Melaka, Malaysia, 358-372.

Ghanim, T. M., Khalil, M. I., & Abbas, H. M. (2020). Comparative study on deep convolution neural networks DCNN-based offline Arabic handwriting recognition. *IEEE Access, 8*, 95465-95482.

Guellil, I., Saâdane, H., Azouaou, F., Gueni, B., & Nouvel, D. (2021). Arabic natural language processing: An overview. *Journal of King Saud University-Computer and Information Sciences, 33(5)*, 497-507. doi: 10.1016/j.jksuci.2019.02.006

Gupta, D., & Bag, S. (2021). CNN-based multilingual handwritten numeral recognition: a fusion-free approach. *Expert Systems with Applications, 165*, 113784.

Haq, M. U., Sethi, M. A. J., & Rehman, A. U. (2023). Capsule Network with Its Limitation, Modification, and Applications—A Survey. *Machine Learning and Knowledge Extraction, 5(3)*, 891-921.

Kavitha, B. R., & Srimathi, C. B. (2022). Benchmarking on offline Handwritten Tamil Character Recognition using convolutional neural networks. *Journal of King Saud University-Computer and Information Sciences, 34(4)*, 1183-1190.

Maalej, R., & Kherallah, M. (2022). New MDLSTM-based designs with data augmentation for offline Arabic handwriting recognition. *Multimedia Tools and Applications, 81(7)*, 10243-10260.

Meddeb, O., Maraoui, M., & Zrigui, M. (2021). Arabic text documents recommendation using joint deep representations learning. *Procedia Computer Science, 192*, 812-821.

Moreno-Barea, F. J., Jerez, J. M., & Franco, L. (2020). Improving classification accuracy using data augmentation on small data sets. *Expert Systems with Applications, 161*, 113696.

Moudgil, A., Singh, S., Gautam, V., Rani, S., & Shah, S. H. (2023). Handwritten devanagari manuscript characters recognition using capsnet. *International Journal of Cognitive Computing in Engineering, 4*, 47-54.

Nahar, K. M., Alsmadi, I., Al Mamlook, R. E., Nasayreh, A., Gharaibeh, H., Almuflih, A. S., & Alasim, F. (2023). Recognition of Arabic Air-Written Letters: Machine Learning, Convolutional Neural Networks, and Optical Character Recognition (OCR) Techniques. *Sensors, 23(23)*, 9475.

Pareek, J., Singhania, D., Kumari, R. R., & Purohit, S. (2020). Gujarati handwritten character recognition from text images. *Procedia Computer Science, 171*, 514-523.

Peng, D., Jin, L., Ma, W., Xie, C., Zhang, H., Zhu, S., & Li, J. (2022). Recognition of handwritten Chinese text by segmentation: a segment-annotation-free approach. *IEEE Transactions on Multimedia, 25*, 2368-2381

Prieto, J. R., Andrés, J., Granell, E., Sánchez, J. A., & Vidal, E. (2023). Information extraction in handwritten historical logbooks. *Pattern Recognition Letters, 172*, 128-136.

Rabi, M., Amrouch, M., & Mahani, Z. (2018). Recognition of cursive Arabic handwritten text using embedded training based on hidden Markov models. *International journal of pattern recognition and Artificial Intelligence, 32(01)*, 1860007.

Shuvo, M. I. R., Akhand, M. A. H., & Siddique, N. (2022). Handwritten numeral recognition through superimposition onto printed form. *Journal of King Saud University-Computer and Information Sciences, 34(9)*, 7751-7764.

Singh, S., Sharma, A., & Chauhan, V. K. (2021). Online handwritten Gurmukhi word recognition using fine-tuned deep convolutional neural network on offline features. *Machine Learning with Applications, 5*, 100037.

Song, X., Cong, Y., Song, Y., Chen, Y., & Liang, P. (2022). A bearing fault diagnosis model based on CNN with wide convolution kernels. *Journal of Ambient Intelligence and Humanized Computing, 13(8)*, 4041-4056.

Souibgui, M. A., Fornés, A., Kessentini, Y., & Megyesi, B. (2022). Few shots are all you need: a progressive learning approach for low resource handwritten text recognition. *Pattern Recognition Letters, 160*, 43-49.

Zhang, T., Chen, J., Li, F., Zhang, K., Lv, H., He, S., & Xu, E. (2022). Intelligent fault diagnosis of machines with small & imbalanced data: A state-of-the-art review and possible extensions. *ISA transactions, 119*, 152-171.