**Communications in Mathematics and Applications**
Vol. 15, No. 5, pp. 1561–1578, 2024
ISSN 0975-8607 (online); 0976-5905 (print)
Published by RGN Publications
DOI: 10.26713/cma.v15i5.3309

http://www.rgnpublications.com

Research Article

# Predicting Dyslexia in Arabic-Speaking Children Through Handwritten Images Using Deep Learning Methods

Norah Dhafer Alqahtani*[1,2], Bander Alzahrani[1], Muhammad Sher Ramzan[1] and Manal Altuwijri[3]

[1] *Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia*
[2] *Department of Information Systems, King Khaled University, Abha 61421, Saudi Arabia*
[3] *Ministry of Education, Riyadh, Al-Arid Education Office, Jeddah, Saudi Arabia*
***Corresponding author:** nhalqahtani@stu.kau.edu.sa

**Received:** September 7, 2024 **Accepted:** December 23, 2024

**Abstract.** Dyslexia is categorized as an educational disorder that affects the ability to spell, read, and write. It is considered one of the most common learning difficulties in Saudi Arabian schools. Research has demonstrated the close connection between the reading and writing processes; moreover, deficiencies or difficulties in an individual's reading process have been associated with difficulties in writing. In this paper, we utilized a novel method to reveal dyslexia among children in elementary school in Saudi Arabia using handwritten images of alphabet letters. We present new datasets of Arabic letters written by children aged 7-10 in early childhood schools in Jeddah, Saudi Arabia. These datasets contain 3,611 characters written by two groups of children: one comprised of children who suffered from dyslexia and one without this disorder. Furthermore, we propose a *Convolutional Neural Network* (CNN) model for the automatic detection of dyslexia using handwriting. The results demonstrate that our model exhibits outstanding performance, attaining accuracy levels of 95.30% and 95.97% before and after the data augmentation technique, respectively, on our dataset.

**Keywords.** Dyslexia, Deep Learning, Dyslexia Prediction, Dyslexia Classification, Handwritten Image, Handwriting Recognition, CNN

**Mathematics Subject Classification (2020).** 68W01

## 1. Introduction

Dyslexia is recognised as a type of learning disability that hinders individuals of normal intelligence and without any sensory impairments in vision or hearing from acquiring basic reading and spelling skills, receiving appropriate education, and accessing socio-cultural opportunities. The nature of dyslexia is occasionally misunderstood, and some specialists have focused on questions such as whether writing problems can signal reading problems or whether anomalies in handwriting indicate writing or fine motor impairments. Nevertheless, extensive research conducted in the field of school psychology has consistently demonstrated that these assumptions are unfounded. It has been established that there is a strong correlation between reading and writing, and students who encounter difficulties in acquiring reading skills often face similar challenges in developing their writing abilities (Spoon *et al*. [23]).

The writing problems experienced by pupils with dyslexia can be largely ascribed to their difficulty in reading and might appear in several forms in their written work, including deficient spelling, illegible handwriting, limited vocabulary, inadequate idea elaboration, and/or deficient organization (Hebert *et al*. [12]). Children diagnosed with dyslexia often exhibit atypical letter formation characterized by the omission of spaces between words, the extension of letters below the baseline, and frequent erasures. Moreover, instructors are frequently unaware of these subtle distinctions.

Germano *et al*. [11] observed that a significant proportion (84.3%) of students diagnosed with dyslexia exhibited characteristics indicative of a deficiency in handwriting skills. However, when teachers were queried regarding the same students, their responses predominantly indicated a perception of the infrequent occurrence of such difficulties. This finding suggests a lack of awareness among teachers regarding the handwriting challenges experienced by these students. Shaywitz [21] who specializes in dyslexia detection, found that mispronouncing common words, not recognizing rhyming patterns, poor penmanship, and messy handwriting are among the most reliable indicators of dyslexia.

Some features of dyslexic handwriting include writing in an upward or downward slant, often known as ascending or descending handwriting, respectively, even when using lined paper. In addition, dyslexic handwriting exhibits malformed curvatures and angles of letters, irregular or no spaces between words, inconsistent sizes of letters, or generally inadequate form (Spoon *et al*. [23]). Additionally, mirror writing is frequently observed in individuals with dyslexia and can serve as a prominent indicator of visual processing difficulties. These difficulties may manifest in challenges related to visual discrimination, where students struggle to discern differences between images, or in visual directionality, where they encounter difficulties in determining the orientation of images. Although some non-dyslexic children reverse letters, they quickly rewrite them correctly. Dyslexic children, on the other hand, often tend to reverse letters without noticing. According to Orton's observations, children diagnosed with dyslexia tend to exhibit a prolonged persistence of mirror-image confusions and reversals compared to children who acquire reading skills in a typical manner (Fischer and Luxembourger [10]). In a study by Lachmann [14], reversal errors were identified as one of the main signs of developmental dyslexia.

*Deep Learning* (DL) methods have been used to diagnose dyslexia in children since 2010 (Alqahtani *et al*. [4]), along with successfully diagnosing other deficiencies and diseases. DL is a specialized domain within the study of machine learning that focuses on the development of algorithms that are modelled after the complex composition and operation of the human brain, sometimes referred to as *Artificial Neural Networks* (ANNs). In this paper, we will utilize handwriting images in the Arabic alphabet as a novel way to predict dyslexia in Arabic-speaking children using the DL model. The following are the contributions made by this paper: (1) It assembled the first dataset of the Arabic alphabet written by dyslexic elementary school children; (2) it used a data augmentation method in collecting data to maximize the training dataset; and (3) the study constructed a CNN model to predict dyslexia in children through a handwriting dataset.

The remaining portion of the paper is structured as follows. The following section examines relevant research that discusses dyslexia in Arabic speakers or the function of DL models in Arabic handwriting recognition in the literature. The approach of this work, from data collection to the assessment of the suggested CNN model, is described in the third section. The study's findings are explained and discussed in the fourth section, and the paper is concluded in the last portion.

## 2. Related Work

### 2.1 Arabic Handwritten Recognition

Because handwritten character recognition is of great importance in many fields, it has recently attracted increased interest from academics. However, this is still a relatively unexplored research subject. DL models have proven their ability in this field, achieving a generally good level of accuracy in the literature. El-Sawy *et al*. [9] conducted comprehensive data collection of handwritten Arabic characters in order to provide training a deep-learning model. Furthermore, they applied optimization techniques to a convolutional neural network (CNN). The proposed CNN model achieved an impressive accuracy rate of 94.9%. AlJarrah *et al*. [3] created a CNN in 2021 so that it could read handwritten Arabic letters. A dataset including 16,800 pictures of handwritten Arabic characters was used to train the CNN model. Six convolutional layers were used in the suggested model, each with distinct functionalities for picture prediction and recognition, as well as three fully linked layers specifically made for recognising characters. Two independent sets of data were processed using the CNN model, one consisting of 40 samples and the other of 256 samples, resulting in an accuracy rate of 97.2%. In 2021, Alrobah *et al*. [5] introduced a hybrid model that integrated a machine learning model with a DL model. The Hijja dataset, as provided by Altwaijry and Al-Turaiki [6], was preprocessed by the researchers. Subsequently, a CNN was developed in order to extract characteristics from the Arabic letter images.

In a 2021 study, Balha *et al*. [7] concentrated on the text segmentation and recognition phases. A variety of solutions were offered for the task of text segmentation. In the recognition stage, a CNN was employed, wherein a total of 14 distinct native CNN designs were put forth. The model underwent training and evaluation with the 54,115 handwritten Arabic letters in the HMBD database. The study involved conducting experiments utilizing indigenous CNN designs, achieving a peak accuracy rate of 91.96%. In addition, the researchers devised a method known as HMB-AHCR-DLGA, which combines *Genetic Algorithm* (GA) and *Transfer Learning* (TL) techniques. This strategy aims to enhance the hyperparameters and training parameters used during the recognition stage. For the latter strategy, pre-trained CNN models (MobileNetV2, VGG16, and VGG19) were employed. The optimal combinations were determined during a series of five experimental iterations. The best known testing accuracy of 92.88% was obtained in the VGG16 experiment using the AdaMax weights optimiser.

## 2.2 Dyslexia in Arabic Language

Arabic, a Semitic language, is predominantly utilized within the Arab world, an expansive region encompassing the regions of the North Africa and Middle East. It functions as the liturgical language of Islam and is the native tongue of more than 330 million individuals, placing it among the most extensively spoken languages globally (Shaalan *et al*. [20]). The Arabic script is composed of a total of 28 letters, which are commonly referred to as "consonantal" letters due to their primary function of representing consonant sounds. The Arabic script uses a right-to-left writing orientation, and a significant characteristic of this script is the alteration of letter shapes based on their location inside a word (initial, medial, or final) or when they are isolated. In addition to these characters, Arabic script additionally includes diacritics and extra symbols to express vowels and other phonetic qualities. Arabic, unlike English and some other languages, does not have uppercase or lowercase letters (Khreisat [13]).

**Table 1.** Arabic alphabet characters

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Alef | أ | Daal | د | Daad | ض | Kaaf | ك |
| aa | ب | Thaal | ذ | Taa' | ط | Laam | ل |
| Taa' | ت | Raa' | ر | Dhaa' | ظ | Meem | م |
| Thaa' | ث | Zayn | ز | Ayn | ع | Noon | ن |
| Jeem | ج | Seen | س | Ghayn | غ | Haa' | هـ |
| Haa' | ح | Sheen | ش | Fa | ف | Wow | و |
| Khaa' | خ | Saad | ص | Qaaf | ق | Yaa' | ي |

The Arabic alphabet comprises letters that can be classified into groups based on their fundamental shapes. Multiple letters exhibit identical shapes; varying numbers of dots positioned either above or below said shapes (e.g., "ت", "ب" or "ث") serve as an essential method for differentiating between these letters. Abu-rabia and Taha [1] grouped the letters according to the number of dots, identifying 15 letters that are marked with dots. Among these, 10 letters are marked with a single dot, three letters are marked with two dots, and two letters are marked with three dots. Table 1 presents a list of isolated Arabic letters.

Arabic-speaking dyslexic children suffer from poor handwriting, especially at the elementary school stage, when they tend to write letters in a disorganized form, as they are unable to produce the letters' appearance in terms of their curves and direction, in addition to not completing them. Moreover, most of these children write the letters in an inverted form. Figure 1 illustrates some letters written by dyslexic children in elementary school.

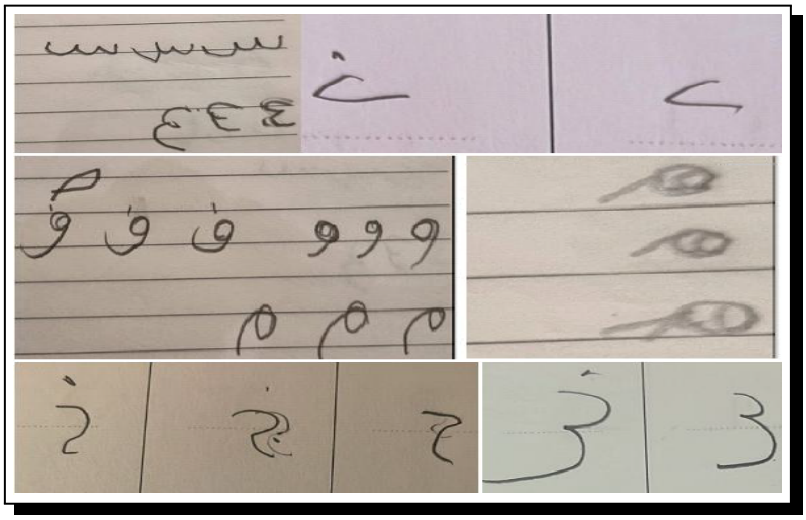

**Figure 1.** Some samples of Arabic characters written by children with dyslexia

When these children undergo early intervention with appropriate tools and software, they can improve their skills and overcome the difficulties they face. This is the motivation behind this research, in which we utilized handwriting images of letters to predict dyslexia in children.

## 3. Proposed System and Methodology

In our research, we built a DL model to predict dyslexia in Arabic-speaking dyslexic children by means of their handwriting images, as illustrated in Figure 2. The following are the steps in the methodology: The first step was Obtaining Images, which was followed by preprocessing of them. Then, the CNN model was constructed, which involved feature extraction and classification. We utilized a data augmentation technique to maximize the training dataset to enhance the model's efficiency. Finally, we evaluated the model's performance based on a confusion matrix.

### 3.1 Obtaining and Digitizing Images

Datasets are essential for the operation of computer vision systems. A benchmark dataset facilitates the expeditious and equitable comparison of various machine learning and DL methodologies and techniques by researchers (Cohen *et al*. [8]).

Despite thorough research, we were unable to find an Arabic handwriting dataset written by dyslexic individuals. Therefore, we decided to collect Arabic handwriting letters written by dyslexic children to create the first Arabic handwriting dataset for dyslexia. We collected this dataset from elementary schools (early childhood schools) in Jeddah, Saudi Arabia. These schools educate children (aged 7-10) from the first to the third grade, including both males and females, as well as children with learning disabilities. Additionally, we collected a control set of handwritten images (non-dyslexic) from the same schools. The criteria for the collected datasets were as follows:

- the control group was free of any diseases or learning difficulties,
- the children with dyslexia were diagnosed by specialist doctors or special education teachers in schools, and
- the two groups ranged in age from 7 to 10 years.

The image collection process was implemented by scanning children's notebooks and exercise papers using a camera and a scanner device and scanning a paper form containing a $4 \times 7$ table that accommodates the writing of 28 letters. The form was distributed to the children to arrange the alphabet's letters in correct order in an isolated format. The digitization stage entailed transforming the input images into an electronic format, which was necessary for the initial stage of preprocessing (Larasati and Keunglam [15]). This scanning yielded 3,611 letters, of which 1,636 images were used as the dyslexic set with the remaining 1,975 serving as the control set.

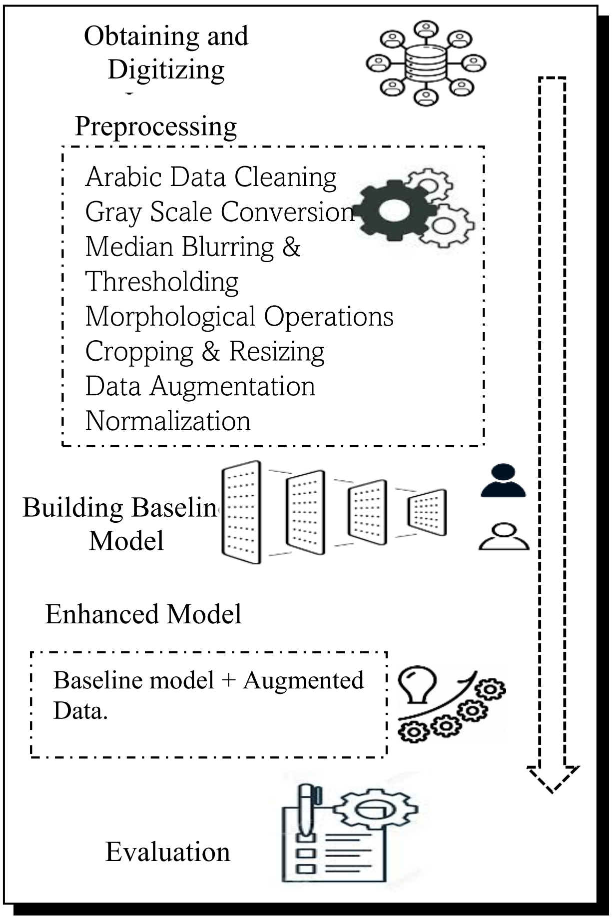


**Figure 2.** Dyslexia Detection in Arabic-speaking children

## 3.2 Data Preprocessing

Prior to putting the data into the model being constructed, it was crucial to undertake the process of data cleaning and normalization in the following manner. The pre-processing stage was mostly targeted at reducing background noise, improving the images' region of interest, and clearly distinguishing the foreground from the backdrop (Madane *et al.* [17]). As we mentioned in the previous step (data collection), the dataset was collected by having the children fill out a form that was distributed to them, as well as by scanning written characters in the children's notebooks. Thus, the preprocessing varied depending on the scanned image. The letters written in the form were passed through different steps than the notebook images, as illustrated in the following:

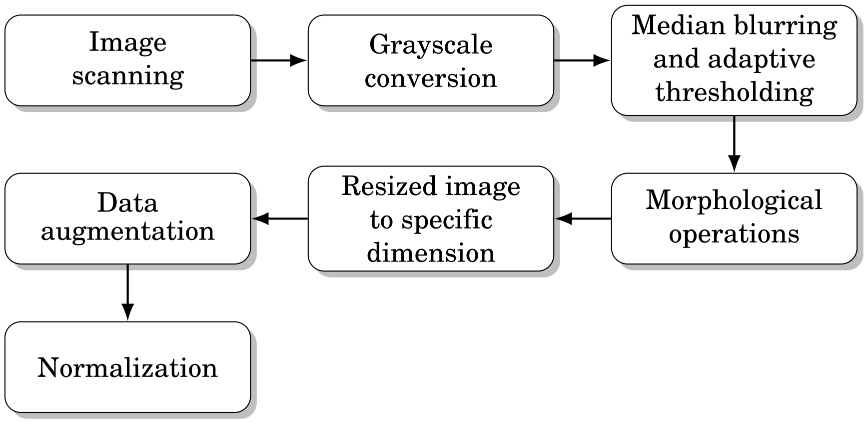


**Figure 3.** Image processing phases

- The first step after scanning the images is to convert the input image into grayscale. This simplifies the image to a single channel, making it easier to process while retaining essential information.
- Median blurring was used to decrease image noise. This is particularly effective in suppressing random variations in pixel intensity. In addition, we applied adaptive thresholding to create a binary image. This adjusts the threshold dynamically based on the local pixel neighborhoods, thus improving adaptability to variations in lighting and contrast.
- Morphological operations (dilation and erosion) were applied to aid in connecting broken parts and smoothing out images. Dilation involves expanding the boundaries of objects in the binary image. This is followed by erosion, which contracts these boundaries. Figure 4 shows how the letter "ض" looks before and after applying these processing operations.

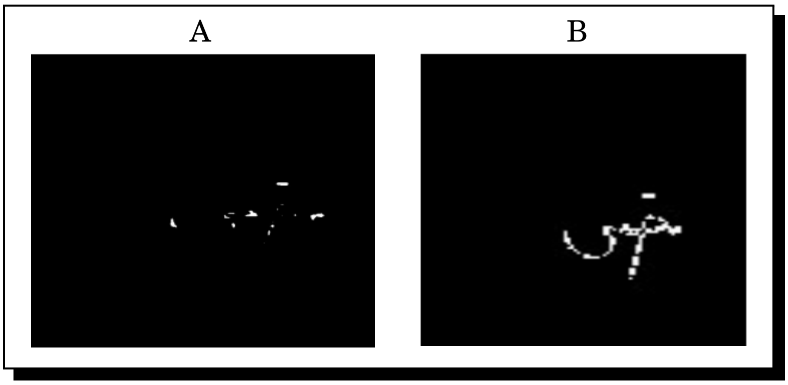


**Figure 4.** The letter "ض" before processing (A) and after preprocessing (B)

- Extraction and resizing: In this step, we identified and extracted individual letters from the binary image and then resized each letter to a fixed size ($32 \times 32$) for standardization.

  Some images scanned from the children's notebooks needed to be cleaned before preprocessing. For these, we utilized Adobe Photoshop software. Figure 5 illustrates this process used on the "ق" character before and after cleaning.

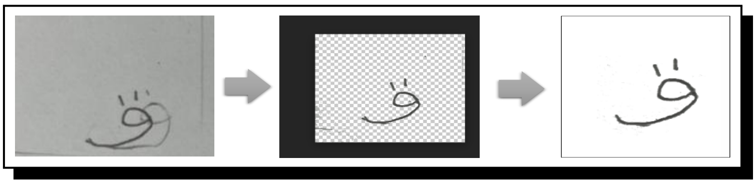

**Figure 5.** Cleaning process for the letter ق

- The entire dataset was finally converted into .CSV data. Figure 6 presents some sample images taken from the dataset.

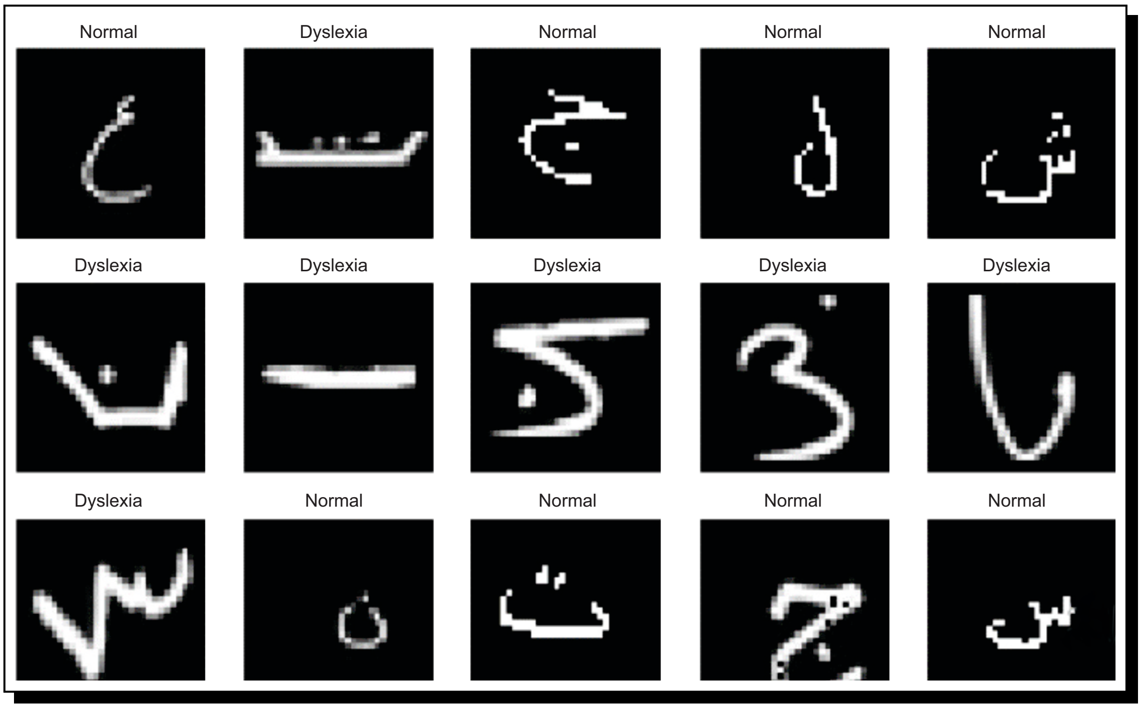


**Figure 6.** Some samples from the study's dataset

## 3.3 Prediction Model

The CNN is widely regarded as the predominant methodology employed in the automated detection of characters and digits within computer systems (Albattah and Albahli [2]). We utilized this model in our research, as it automatically detects and extracts robust and intricate features.

The CNN is a computational model designed to emulate the human brain's cognitive processes in the context of picture perception, identification, and comprehension. The procedure involves analyzing pixel data to detect distinctive traits that can be used as points of identification (Albattah and Albahli [2]). It functions by allocating significance, in the form of learnable biases and weights, to specific components of input images (Albattah and Albahli [2]). This process enables differentiation between these images, ultimately facilitating the recognition of their content. The basic components of this model are the convolutional layer, activation layer, pooling layer, and fully connected layer.

### 3.3.1 Structure of the Proposed CNN Model

In our study, we first build a CNN model as a baseline model for prediction. We try to enhance the model through data augmentation techniques. As we see in Figure 7, our model has four hidden layers, an input layer, and a fully connected layer for classification. Every hidden layer includes a convolutional layer with ReLU, batch normalization, and max pooling, in addition to the dropout layers. The layers of classification include a flatten layer, three fully connected layers, and a batch normalization layer. Each layer was built for a specific task, as explained below.

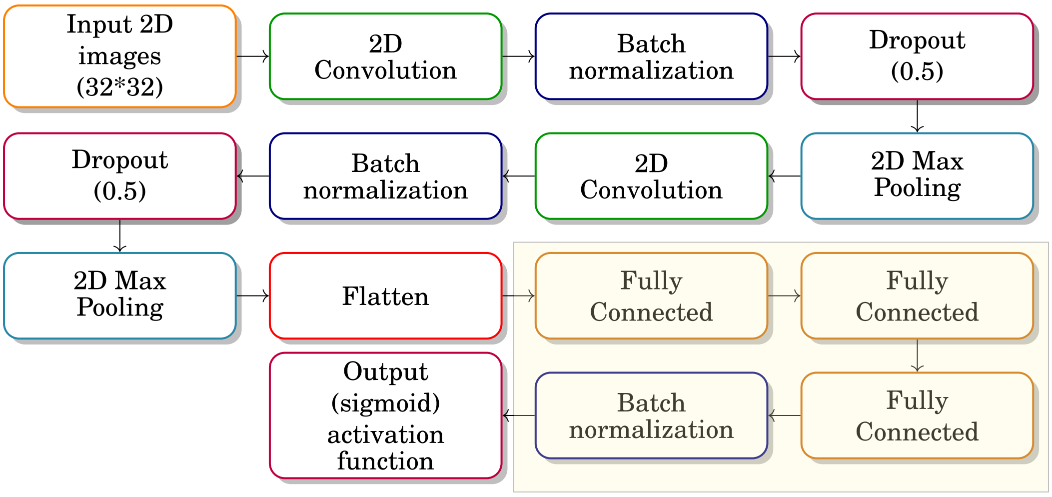


**Figure 7.** CNN Architecture

- *Input layer:* We provide an image of a handwritten number as input. This grayscale digit measures $32 \times 32 \times 1$ pixels (width $\times$ height $\times$ channel). Pixel intensity values range from 0 to 255 (black to white), with each number in between representing a different pixel brightness level.
- *Hidden layer:* The concealed layer is made up of a series of layers that reveal the distinguishing properties of the handwriting digits.
  - *Convolution layer:* This layer has a *kernel*, also known as a *filter*, which is a two-dimensional array of weights. The primary function of this layer is to perform feature extraction, resulting in the generation of a feature map. By performing convolution procedures on specific regions of the initial feature maps, output feature maps are generated. During the process of convolution, kernels acquire knowledge about the characteristics of the handwritten images for two classes and generate more precise categorization. As depicted in (3.1), every characteristic is multiplied by the appropriate value in the kernel. The outcome of the multiplication operation is referred to as the *weight*. Subsequently, we incorporate the bias term and subsequently employ the following non-linear function, specifically the ReLU:

$$F = \sigma(wa + b), \tag{3.1}$$

    where $w$ refers to weight, $a$ to the input feature, $\sigma$ to the activation function, and $b$ denotes bias. This technique is iterated until a final feature map is obtained. This formulation encompasses all fully linked and convolutional layers in this work. Due to the nonlinear nature of a handwritten digit database, it is crucial to incorporate nonlinearity into the CNN. To achieve this, the ReLU is applied after every convolution operation. Utilizing the ReLU function following a convolution operation enhances the performance of the CNN.

- *Rectified Linear Unit (ReLU):* As the convolution layer's output is treated as a linear process, it is crucial to apply a nonlinear function that accurately depicts the behavior of a real neuron after the convolution operation. The activation function commonly employed to model a neuron's output $f$ with respect to its input $x$ is the sigmoid function. Nevertheless, when considering the training period with gradient descent, this activation function exhibits significantly slower performance than the ReLU. Thus, in this study, we employ the ReLU to introduce non-linearity into our neural network (Luo *et al*. [16]). As shown in (3.2), in the case that the input is negative, the ReLU provides an output of 0. Conversely, the ReLU function outputs the input value $x$ if it is positive:

$$f(x) = \begin{cases} x, & x \geq 0, \\ 0, & x < 0. \end{cases} \tag{3.2}$$

- *Batch Normalization (Batch Norm):* This is the most utilized technique in the domain of DL to enhance the speed of learning and develop the training performance of CNNs (Zhang *et al*. [25]). Within the majority of neural networks, the phenomena of exploding or vanishing can potentially occur when employing a learning rate that is excessively high. To tackle these issues, we implemented batch normalization. In addition, batch normalization enhances learning efficiency and serves as a form of regularization to prevent overfitting of the model. By incorporating batch norm and ReLU activation, our method gained outstanding classification accuracy. Batch norm significantly enhanced the efficiency of our model by accelerating the training process, decreasing the time required for both training and testing, and minimizing the susceptibility to initialization variations.
- *Pooling Layer:* The pooling layer reduces certain parameters in the input image resulting in dimensionality reduction. Similar to the filter, it sifts through the entire input. However, unlike the convolutional layer, it is weightless. Instead, the image is subjected to an aggregation function to fill the output array. Typically, two kinds of pooling are employed.
  - *Max Pooling*: This selects the highest value from the portion of the feature map covered by the filter (Nurjannah *et al*. [18]). As a result, the resulting feature map contains the most significant properties of the previous feature map.
  - *Average Pooling*: Average pooling finds the average value of all the elements in the feature map area that the filter covers. Max pooling takes out the most important feature from a certain area of the captured feature map, while average pooling finds the mean of all the features in that area.

We implemented max pooling in our study. Thus, only the crucial features are propagated to the subsequent layer, while the nonessential features are discarded. Our model employs a $2 \times 2$ max-pooling layer kernel along the input spatial dimensions of input, with a stride of 2. The integration of this layer with previous layers allows our model to extract local characteristics effectively and acquire broader global information from the input image.

- *Dropout:* The dropout layer functions as a mask, randomly turning off certain neurons on the following layer while preserving the functionality of all other neurons throughout the training phase. Moreover, the act of randomly eliminating neurons serves to compel the neural network to acquire distributed properties (Yahya *et al*. [24]). Dropout is commonly employed in large networks to enhance generalization and address the issue of overfitting, which can result in intricate co-adaptations on the data used for training (Pham *et al*. [19]). Furthermore, dropout enhances the robustness of the neural network and reduces its reliance on particular quirks present in the training data. Implementing dropout in both hidden and apparent layers typically yields superior outcomes compared to applying dropout alone to a single hidden layer. Therefore, in our model, we randomly eliminated neurons from the neural network's feature extraction layers with a probability of 0.5.

All the previous layers are responsible for feature extraction from handwritten images in the model. The following layers serve classification purposes:

- *Classification layer*
  - *Flatten Layer:* This layer is used to convert the output from the max-pooling layer, the feature map, into a format that the fully connected layers can comprehend. A feature map is a multidimensional array that stores pixel values, but the fully connected layers demand a one-dimensional array as input for computation.
  - *Fully Connected Layer:* This layer handles recognition and detection. Each neuron or node in the previous layer is linked to every neuron in the current layer. The term "fully connected" is used to describe the presence of total connectivity. After training, the fully connected layer's feature vector is utilized to further classify images into multiple categories. In our model, the sigmoid function is used as the activation function in the fully connected layer, which is applied to the training data. This function divides the learnt features from antecedent layers into discrete groups and presents the probability associated with each class, utilizing a threshold of 0.5. Table 2 displays the parameters of our proposed model.

## 4. Experiment and Results

This section provides a detailed description for the setup of current study, encompassing the software requirements.

• *Software*: Jupyter was utilized under the Anaconda package manager to create the CNN model. Python version 3.7 is the programming language utilized.

• *Dataset and model*: The collected dataset was split into two parts: 80% was used for training and 20% was used for testing. The proposed model, explained in the above section, was created as a baseline model for this study. The model was trained using 5-fold cross-validation with a batch size of 32 over 60 epochs. Table 2 presents specific information regarding the training parameters used in the proposed model with the Adam optimizer, which aids in adjusting the updated model's parameters, such as weights, to effectively minimize the loss function.

**Table 2.** Number of parameters of the CNN architecture

| Layer (type) | Output Shape | Param # |
|---|---|---|
| Conv2d-1 | [-1, 8, 30, 30] | 80 |
| LeakyReLU-2 | [-1, 8, 30, 30] | 0 |
| BatchNorm2d-3 | [-1, 8, 30, 30] | 16 |
| Dropout-4 | [-1, 8, 30, 30] | 0 |
| MaxPool2d-5 | [-1, 8, 15, 15] | 0 |
| Conv2d-6 | [-1, 16, 13, 13] | 1,168 |
| LeakyReLU-7 | [-1, 16, 13, 13] | 0 |
| BatchNorm2d-8 | [-1, 16, 13, 13] | 32 |
| Dropout-9 | [-1, 16, 13, 13] | 0 |
| MaxPool2d-10 | [-1, 16, 6, 6] | 0 |
| Flatten-11 | [-1, 576] | 0 |
| Linear-12 | [-1, 160] | 92,320 |
| Linear-13 | [-1, 92] | 14,812 |
| BatchNorm1d-14 | [-1, 92] | 184 |
| Linear-15 | [-1, 2] | 186 |

Total params: 108,798
Trainable params: 108,798
Non-trainable params: 0

Input size (MB): 0.00
Forward/backward pass size (MB): 0.33
Params size (MB): 0.42
Estimated Total Size (MB): 0.75

The accuracy and loss function in the training phase demonstrate our model's performance, where accuracy is a measure of the number of right predictions of the anticipated value matching the true value, while the loss function is the total number of mistakes created for every single sample in the datasets used for training or validation. Figure 8 displays the training and loss curves, which demonstrate the learning behavior of the suggested technique.

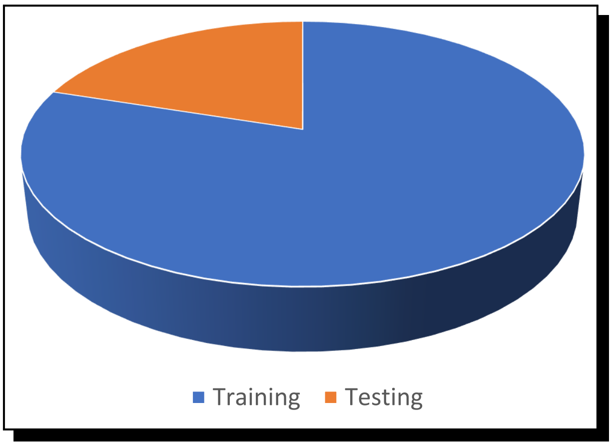


**Figure 8.** Dataset distributions

**Table 3.** Important training parameters in the model

| Model parameters | Value |
|---|---|
| Batch size | 32 |
| Epochs | 64 |
| Learning rate | 0.001 |

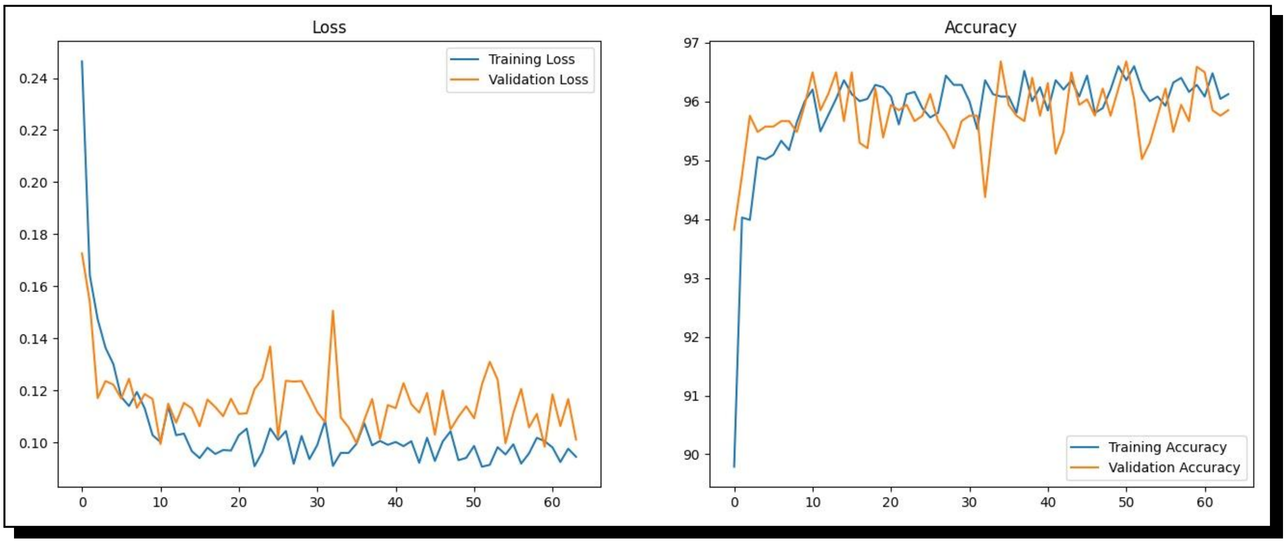


**Figure 9.** Loss and accuracy of the proposed model

The losses of the training and testing model are 0.11 and 0.12, respectively. In addition, to express numerically the classification accuracy, we utilized a confusion matrix, which is the approach most frequently employed in DL and machine learning for this purpose. It contains details on the real and expected classes that a classification system has identified. The confusion matrix consists of two dimensions: one representing the actual classes and the other representing the expected classes. Each row in the table corresponds to a specific class example, while each column shows the anticipated state of a class. The confusion matrix has four elements: TP (true positive), TN (true negative), FP (false positive), and FN (false negative).

• The model's accuracy is measured using the accuracy metric, which is determined by dividing the total amount of data by the number of correctly classified data.

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{EP}+\text{FN}}. \tag{4.1}$$

1. *Precision*: Precision measures the accuracy of correctly predicting positive data. Put simply, a high level of precision results in a reduced number of false positives.

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}. \tag{4.2}$$

2. *Recall*: Recall is a metric used to measure the classifier's ability to correctly identify all relevant instances. Greater recall signifies a reduced number of false negatives, whereas lesser recall signifies an increased number of false negatives. Enhancing recall frequently leads to a reduction in precision.

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}. \tag{4.3}$$

3. The precision and recall products are divided by their sum to determine the F1-score.

$$\text{F1-Score} = \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} * 2. \tag{4.4}$$

Figure 10 displays the confusion matrix, which demonstrates the classification model's efficacy, with 95.75 training accuracy and 95.30 testing accuracy for the base model.

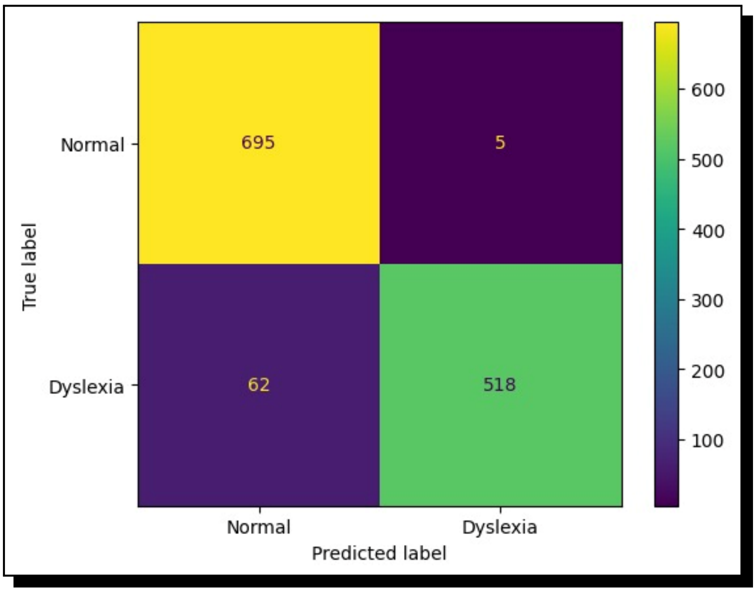


**Figure 10.** Confusion matrix of the baseline model

• *Data Augmentation*: DL models have exhibited exceptional performance on numerous computer vision tasks (Larasati and Keunglam [15]). However, neural networks rely heavily on huge datasets to reduce the risk of overfitting. Overfitting is a phenomenon characterized by the excessive flexibility of a neural network, resulting in the ability to accurately represent the training data without generalizing to new, unseen data with minimal error (Madane *et al*. [17]). Our dataset needs to be maximized to enhance the efficiency of the DL model, which requires a large-scale dataset for training purposes. Therefore, we applied augmentation techniques to 70% of the dataset for each class, and 30% were kept for testing. A further technique applied in this study was Gaussian blur.

The process of Gaussian blur involves modifying the intensity of each pixel in a picture based on the Gaussian distribution function. This is achieved by calculating the weighted average value of the pixel and its neighboring pixels (Song *et al*. [22]):

$$GS(a,b) = \frac{1}{2\pi\sigma^2} e^{-(+a^2+b^2/2\sigma^2)}, \tag{4.5}$$

where $a$ and $b$ refer to the pixels' coordinates in the Gaussian blur kernel, and $\sigma$ denotes the standard deviation of the distribution. The implementation of Gaussian blur in this experiment utilized PyTorch's Gaussian blur function. The input parameters consist of the kernel size and a range of potential values for the standard deviation $\sigma$. In this investigation, the kernel size was defined as $3 \times 3$, and the standard deviation was specified as between the range of 0.1–2.0.

After applying the Gaussian blur technique, the size of the training data was maximized from 2,527 to 5,054 images. The accuracy after applying this technique reached 96.863. Figure 11 illustrates the training and loss curves, which highlight the model's learning behaviour after using the suggested data augmentation strategy. The model's training loss after data augmentation is 0.09, whereas the testing loss is 0.10.

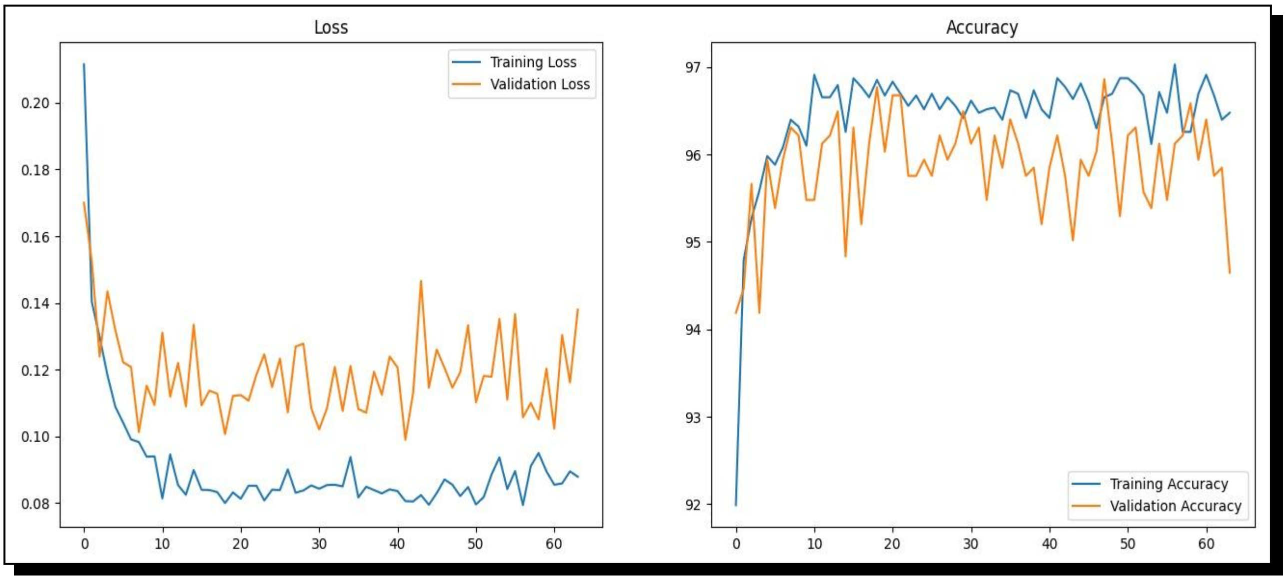


**Figure 11.** Loss and accuracy of the proposed model

Figure 12 displays the modified model's confusion matrix, which clarifies the slight difference from the baseline classification report where the enhanced model reached 96.52 in training and 95.97 in testing.

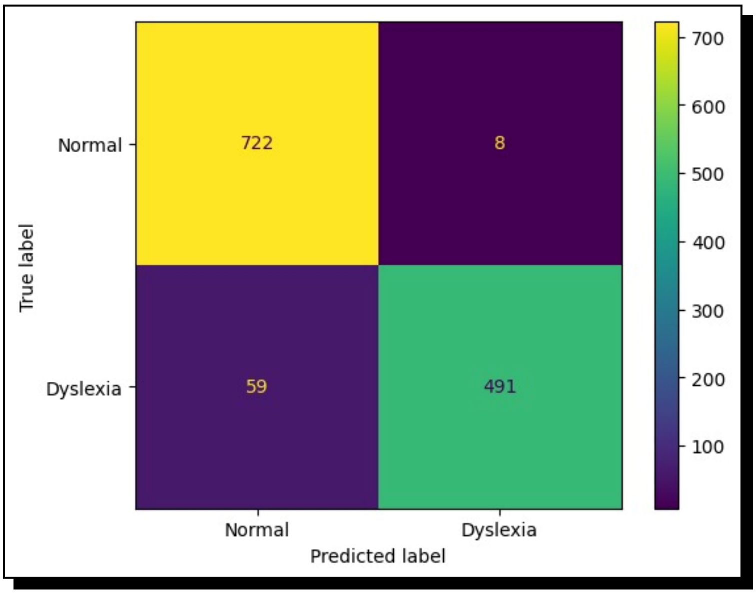


**Figure 12.** Confusion matrix of the enhanced model

As per above results, the accuracy of each model, baseline and enhanced, increased by 0.67%. The total amount of time spent on training and testing for each model is illustrated in Table 4.

**Table 4.** Training and testing time for proposed models

| Model | Total training time | Total testing time |
|---|---|---|
| Baseline model | 43.1 s | 281 ms |
| Enhanced model | 1 min 11 s | 279 ms |

The baseline model was faster in the training phase than the enhanced model (utilizing data augmentation) due to the increase in the data in the training dataset. The model needed only 43.1 s for training, while the enhanced model after data augmentation needed 1 min 11 s.

## 5. Conclusion

This paper aimed to find a new way to detect dyslexia symptoms among Arabic-speaking children in elementary school through photos of their handwriting. Handwritten pictures were collected from two groups (non-dyslexic and dyslexic) in early childhood schools in Jeddah. The study proposed detecting dyslexia by using a DL model (a CNN), which is an automated method that assists in overcoming the time-consumption issues that occur when using other methods for the detection of this disorder. By using Python, we created a CNN model with two convolution layers, followed by activation layers and max pooling layers. The model was supported by batch normalization and dropout layers to aid in increasing its accuracy and accelerating the training process, as well as overcoming overfitting problems. The model succeeded in predicting dyslexia symptoms and achieved an accuracy that reached 95.97%. The Gaussian blur technique was applied to the training datasets to maximize the datasets, slightly enhancing the model's performance. This study is considered the first to predict symptoms of dyslexia through images of handwriting in Arabic-speaking children; therefore, it required pioneering efforts in collecting data related to dyslexic children in the data collection phase.

## Acknowledgments

We deeply thank the specialist teachers and all the children (dyslexic and not) for collaborating with us, filling out the forms, and providing their notebooks for scanning Arabic letters.

This research work was funded by Institutional Fund Projects under grant No. (IFPRC-211-166-2020). Therefore, authors gratefully acknowledge technical and financial support from the Ministry of Education and King Abdulaziz University, Jeddah, Saudi Arabia.

### Competing Interests

The authors declare that they have no competing interests.

### Authors' Contributions

All the authors contributed significantly in writing this article. The authors read and approved the final manuscript.

## References


[1] S. Abu-rabia and H. Taha, Reading and spelling error analysis of native, *Reading and Writing* **17**(7-8) (2004), 651 – 690, DOI: 10.1007/s11145-004-2657-x.

[2] W. Albattah and S. Albahli, Intelligent Arabic handwriting recognition using different standalone and hybrid CNN architectures, *Applied Sciences* **12**(19) (2022), 10155, DOI: 10.3390/app121910155.

[3] M. N. AlJarrah, M. M. Zyout and R. Duwairi, Arabic handwritten characters recognition using convolutional neural network, in: *2021 12th International Conference on Information and Communication Systems* (ICICS2021, Valencia, Spain, 2021), pp. 182 – 188 (2021), DOI: 10.1109/ICICS52457.2021.9464596.

[4] N. D. Alqahtani, B. Alzahrani and M. S. Ramzan, Deep learning applications for dyslexia prediction, *Applied Sciences* **13**(5) (2023), 2804, DOI: 10.3390/app13052804.

[5] N. Alrobah and S. Albahli, A hybrid deep model for recognizing Arabic handwritten characters, *IEEE Access* **9** (2021), 87058 – 87069, DOI: 10.1109/ACCESS.2021.3087647.

[6] N. Altwaijry and I. Al-Turaiki, Arabic handwriting recognition system using convolutional neural network, *Neural Computing and Applications* **33**(7) (2021), 2249 – 2261, DOI: 10.1007/s00521-020-05070-8.

[7] H. M. Balaha, H. A. Ali, E. K. Youssef, A. E. Elsayed, R. A. Samak, M. S. Abdelhaleem, M. M. Tolba, M. R. Shehata, M. R. Mahmoud, M. M. Abdelhameed and M. M. Mohammed, Recognizing arabic handwritten characters using deep learning and genetic algorithms, *Multimedia Tools and Applications* **80**(21-23) (2021), 32473 – 32509, DOI: 10.1007/s11042-021-11185-4.

[8] G. Cohen, S. Afshar, J. Tapson and A. Van Schaik, EMNIST: Extending MNIST to handwritten letters, in: *Proceedings of the 2017 International Joint Conference on Neural Networks* (IJCNN, Anchorage, AK, USA, 2017), pp. 2921 – 2926, (2017), DOI: 10.1109/IJCNN.2017.7966217.

[9] A. El-Sawy, H. EL-Bakry and M. Loey, CNN for handwritten Arabic digits recognition based on LeNet-5, in: *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics* (AISI 2016, Advances in Intelligent Systems and Computing), A. Hassanien, K. Shaalan, T. Gaber, A. Azar and M. Tolba (editors), Vol. 533, Springer, Cham., (2017), DOI: 10.1007/978-3-319-48308-5_54.

[10] J.-P. Fischer and C. Luxembourger, Commentary: Mirror-image equivalence and interhemispheric mirror-image reversal, *Frontiers in Human Neuroscience* **12** (2018), 140, DOI: 10.3389/fnhum.2018.00375.

[11] G. D. Germano, C. Giaconi and S. Capellini, Characterization of Brazilians students with dyslexia in handwriting proficiency screening questionnaire and handwriting scale, *Psychology Research* **6**(10) (2016), 590 – 597, DOI: 10.17265/2159-5542/2016.10.004.

[12] M. Hebert, D. M. Kearns, J. B. Hayes, P. Bazis and S. Cooper, Why children with dyslexia struggle with writing and how to help them, *Language, Speech, and Hearing Services in Schools* **49**(4) (2018), 843 – 863, DOI: 10.1044/2018_LSHSS-DYSLC-18-002.

[13] L. Khreisat, A machine learning approach for Arabic text classification using $N$-gram frequency statistics, *Journal of Informetrics* **3**(1) (2009), 72 – 77, 2009, DOI: 10.1016/j.joi.2008.11.005.

[14] T. Lachmann, Experimental approaches to specific disabilities in learning to read: The case of Symmetry Generalization in developmental dyslexia, in: *Advances in Cognitive Science*, N. A. Srinivasan, K. Gupta and J. Pandey, Sage, Thousand Oaks, CA, pp. 321 – 342 (2008).

**[15]** R. Larasati and H. Keunglam, Handwritten digits recognition using ensemble neural networks and ensemble decision tree, in: *Proceedings of the 2017 International Conference on Smart Cities, Automation & Intelligent Computing Systems* (ICON-SONICS Yogyakarta, Indonesia, 2017), pp. 99 – 104, (2017), DOI: 10.1109/ICON-SONICS.2017.8267829.

**[16]** W. Luo, Y. Li, R. Urtasun and R. Zemel, Understanding the effective receptive field in deep convolutional neural networks, in: *Proceedings of the 30th Conference on Neural Information Processing Systems* (NIPS'16, Barcelona, Spain), Curran Associates Inc., Red Hook, NY, USA, pp. 4905 – 4913, (2016), DOI: 10.5555/3157382.3157645.

**[17]** V. Madane, K. Ovhal and M. Bhong, Handwriting recognition using artificial intelligence neural network and image processing, *International Research Journal of Modernization in Engineering Technology and Science* **5**(3) (2023), 1432 – 1447, DOI: 10.56726/irjmets34395.

**[18]** A. F. Nurjannah, A. S. D. Kurniasari, Z. Sari and Y. Azhar, Pneumonia image classification using CNN with max pooling and average pooling, *Jurnal RESTI* (*Rekayasa Sistem dan Teknologi Informasi*) **5**(2) (2021), 330 – 338, DOI: 10.29207/resti.v6i2.4001.

**[19]** V. Pham, T. Bluche, C. Kermorvant and J. Louradour, Dropout improves recurrent neural networks for handwriting recognition, in: *Proceedings of the 14th International Conference on Frontiers in Handwriting Recognition* (Greece, 2014, pp. 285 – 290), (2014), DOI: 10.1109/ICFHR.2014.55.

**[20]** K. Shaalan, S. Siddiqui, M. Alkhatib and A. A. Monem, Chapter 3: Challenges in Arabic natural language processing, in: *Computational Linguistics, Speech and Image Processing for Arabic Language*, pp. 59 – 83 2018, DOI: 10.1142/9789813229396_0003.

**[21]** S. Shaywitz, Overcoming dyslexia: A new and complete science-based program for reading problems at any level, 1st edition, Alfred A. Knopf, Inc., New York, USA, ix + 414 pages (2003).

**[22]** C. Song, W. Xu, Z. Wang, S. Yu, P. Zeng and Z. Ju, Analysis on the impact of data augmentation on target recognition for UAV-based transmission line inspection, *Complexity* **2020**(1) (2020), 3107450, DOI: 10.1155/2020/3107450.

**[23]** K. Spoon, K. Siek, D. Crandall and M. Fillmore, Can we (and should we) use AI to detect dyslexia in children's handwriting?, in: *Proceedings of the AI for Social Good workshop at NeurIPS* (2019), Vancouver, Canada, pp. 1 – 6, (2019).

**[24]** A. A. Yahya, J. Tan and M. Hu, A novel handwritten digit classification system based on convolutional neural network approach, *Sensors* **21**(18) (2021), 6273, DOI: 10.3390/s21186273.

**[25]** K. Zhang, W. Zuo, Y. Chen, D. Meng and L. Zhang, Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising, *IEEE Transactions on Image Processing* **26**(7) (2017), 3142 – 3155, DOI: 10.1109/TIP.2017.2662206.