International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Deciphering Excellence: Comparative Study of Deep Learning Models in Handwriting Recognition

Sunitha S. Nair*[1], P. Ranjit Jeba Thangaiah[2]

Submitted: 08/12/2023 **Revised**: 15/01/2024 **Accepted**: 29/01/2024

Abstract: In the digital age, handwritten character recognition (HCR) is a vital technique that makes handwritten text easier to read and use by converting it into a machine-encoded format. Handwritten words, symbols, or characters can be automatically recognized and interpreted using HCR. The main objective of HCR is to transform handwritten text into machine-encoded text and make it readable and usable in digital formats. This technology falls under the category of optical character recognition (OCR), which is a broad term that includes handwritten and printed text recognition. This paper presents an efficient HCR model that makes use of Deep Convolutional Neural Networks (DCNN) with variable filters and feature fusion. The model's performance is gauged using a handcrafted dataset, and comparative analysis is conducted with CNN+ Bi-LSTM and CNN+ Bi-GRU models. The suggested Deep CNN with Variable Filters and Feature Fusion Model surpasses the other models with an accuracy of 98.86%. The findings underscore the effectiveness of merging sequential and convolutional modelling approaches and stress the need for complex architectures to achieve the best accuracy possible in HCR challenges. This study advances the field and emphasizes the importance of technological innovation in enhancing data input techniques across businesses, digitizing manuscripts, and protecting cultural heritage.

Keywords: *Convolutional Neural Network, Deep Learning, Handwritten Character Recognition, Optical Character Recognition, Pattern Recognition, Recurrent Neural Network*

1. Introduction

Handwriting is a time-honoured and traditional form of communication that is always evolving as a result of cultural and technical advances. Handwriting has three fundamental properties. It is made up of synthetic graphical marks applied to a surface with the intention of communicating a message, which is made possible by the mark's traditional association with language [1]. Writing was first used by humans as a tool for communication and as a way to convey emotions long ago. Many aspects of civilization and culture are thought to have been made possible by writing. With the invention of writing, people could now preserve a wide range of information, including events, history, literature, law, science, and mathematics. Writing is sometimes described as a formalized system of standard symbols used to represent thoughts according to a set of rules [2]. The development of languages, scripts, and alphabets was a result of cultural and civilizational advancements that aimed to improve communication and fact-recording. Each person has their own penmanship or handwriting, and each has their own writing style. It is also influenced by other factors, such as the writer's mood, writing medium, surroundings, etc. Handwriting is still prevalent in the era of digital computers because paper and pens are more convenient than keyboards in many situations [3]. Figure 1 shows a few examples of handwritten images.

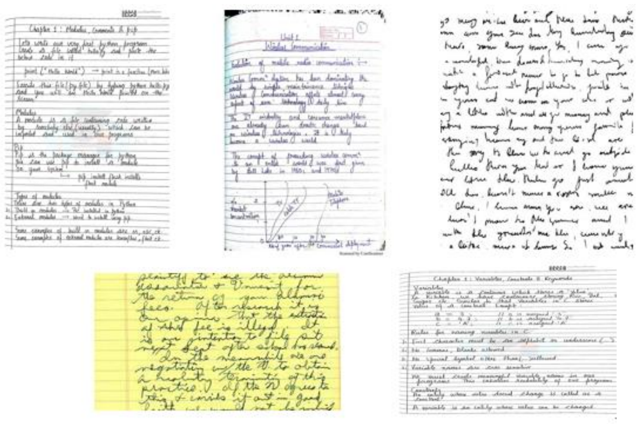


Fig. 1. Illustration of Handwritten Characters

1.1 Handwritten Character Recognition

There are many practical and commercial uses for automatic handwritten information recognition in various places [4]. Automatic reading framework can save a lot of work, even if they can only recognize half of the documents that must be processed daily in these kinds of businesses. The need for HCR increased along with the introduction of gadgets like digital computers and smartphones that could successfully reinterpret and digitize the data entered. In HCR, digital text is generated by interpreting scanned input from documents,

[1,2]Department of Digital Sciences
[1]Research Scholar, Karunya Institute of Technology and Science, Coimbatore, Tamil Nadu, India
[2]Associate Professor, Karunya Institute of Technology and Science, Coimbatore, Tamil Nadu, India
E-mail Id: [1] sunithasnair24@gmail.com
[2]jebathangaiahranjit@gmail.com
** Corresponding Author Email: sunithasnair24@gmail.com*

images, and real-time devices such as tablets, tabloids, digitizers, etc. The automated extraction of textual data from document images is a crucial aspect of document image analysis. Tshis is accomplished by employing the OCR tool, which describes the procedure of transferring optically acquired images of handwritten or machine-printed text (letters, numbers, and symbols) into a format that can be edited on a computer [5].

1.2 Classification of Handwritten Character Recognition

Depending on how the data is acquired, HCR can be categorized into two types:

- Online Character Recognition
- Offline Character Recognition

Online HCR is the ability to recognize handwriting that has been scanned and digitally stored using various techniques [6]. Typically, an electronic surface is used in conjunction with a specialized pen. The 2-D coordinates of succeeding locations are measured as a function of time and are recorded in order as the pen grows across the surface. Data that is obtained online is less noisy and requires less processing. Fast algorithms are needed for skew correction, normalization, etc. Pen trajectory data facilitates and expedites feature extraction procedures, including the detection of strokes, stroke orientations, corners, loops, cusps, etc. [7]. The process of recognizing characters from the movement of the pen tip while writing is called online recognition. The computer incorporates a variety of advanced input techniques, such as hand-held devices and digitizers, as a result of advancements in the digital industry. These portable gadgets are bringing with them a little pen-based instrument that serves as an input interface. These handheld gadgets are convenient in terms of price, mobility, and usefulness. This input device, which includes Personal Digital Assistants (PDAs), is an electronic pen model designed to provide data input on a system screen. Handheld devices, such as electronic or tablet digitizers, are used as transducers to gather handwritten data for online systems that use handwriting recognition.

In contrast, offline handwriting recognition is the process of identifying words that have been digitally saved in grayscale format after being scanned from a surface (such as a sheet of paper) [8]. In order to enable better recognition, further processing is typically carried out after storage. The quality of the writer, paper, pen, ink, and scanner, among other aspects, might cause noise in the input image, which can significantly impact recognition. As a result of the lack of temporal information, stroke level processing requires intricate procedures to retrieve strokes from the image; yet, the sequence in which these strokes are formed is not reversible. Since the entire image is accessible, these systems can also extract offline features such as positional, geometrical, and statistical information, among others. Online character recognition differs greatly from offline character recognition in that the former uses real-time contextual information, while the latter does not. The processing architectures and techniques diverge significantly as a result of this difference. Before digitization, the entire document needs to be written by hand using a pen on paper and scanned using a scanner [9]. The written record was originally a bitmap image before the digital form was recreated. The processes of identifying text, detecting lines from a text document, identifying words from detected lines, idesntifying characters from words, and categorizing that discovered character are all included in the offline document recognition procedures. The conversion of a physically written image copy into an electronic-type file is the primary use case for offline recognition. Figure 2 shows the online and offline HCR processing.

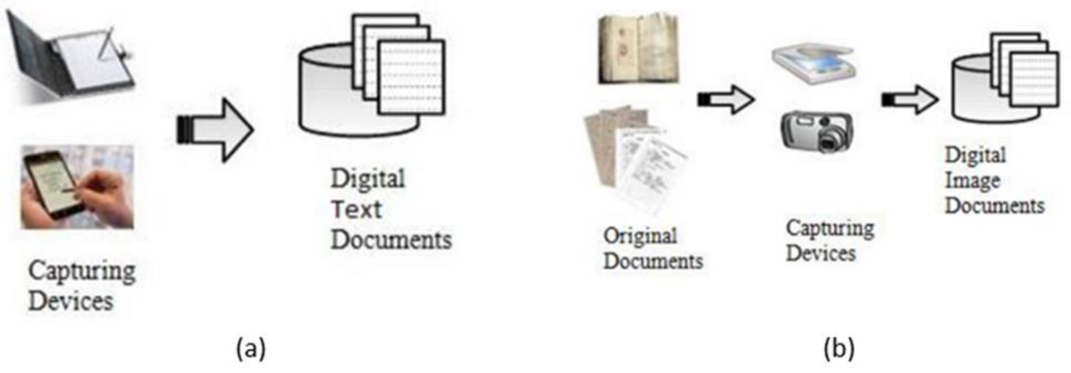


Fig.2. (a) On-line handwriting capturing process (b) Digitization process of offline documents

1.3 Deep Learning (DL) in Handwritten Character Recognition

The field of HCR has undergone a revolution due to DL, which has produced potent tools for automated handwritten text interpretation. Accurate recognition was difficult because traditional approaches frequently battled with the inherent variety in human handwriting. Handwritten characters include intricate patterns and hierarchical representations that can be captured with amazing success by DL, specifically using CNNs and RNNs. DL models perform exceptionally well in feature extraction and hierarchical learning for HCR, allowing them to automatically extract pertinent characteristics from unprocessed input data. CNNs are ideally suited for image-based tasks because they are skilled at collecting spatial hierarchies in pixel patterns [10]. However, RNNs are useful for identifying the temporal connections in handwritten strokes since they are good at processing sequential data [11]. Advanced performance in identifying intricate and varied handwritten characters has been achieved through the amalgamation of different designs, occasionally in hybrid forms. DL has therefore been essential in improving the precision and effectiveness of HCR systems, which have a variety of uses, including automated data entry and document digitization.

The major contribution of this paper includes:

- Development of an efficient framework for offline HCR using deep CNN with variable filters and future fusion.
- Utilization of large dataset for improves the accuracy of HCR.
- Performance comparison of the suggested method with CNN- LSTM and CNN- Bi- GRU hybrid models for HCR.

The rest of the paper is arranged as follows: A review of existing works is given in Section 2, identifying topics that need more investigation. A detailed explanation of the process is provided in Section 3. The results of the suggested strategy are thoroughly discussed in Section 4, along with a comparison of its performance to those of alternative approaches. Finally, the study wraps up in Section Five with a summary of the results.

2 Literature Review

Ritik Dixit et al. [12] developed three models based on deep and machine learning methods for handwritten digit recognition utilizing MNIST datasets. Handwritten digit recognition is achieved by the use of CNN models, multi-layer perceptrons, and support vector machines. According to the simulation findings, CNN provided the most accurate results for recognizing handwritten digits. A DL approach for handwritten character identification was developed by Khandokar et al. [13]. This work primarily aims to investigate the capacity of CNN to effectively recognize characters from the NIST dataset. It was identified that when the training image count increased, the accuracy measured from 200 training images, which was 65.32%, improved progressively. An HCR model utilizing SVM, DT, RF, ANN, KNN, and K-means algorithms was proposed by Rabia Karakaya and Serap Kazan [14]. The efficacy of multiple algorithms on the same database was evaluated, and the operational logic of the handwriting digit recognition process was examined. High-accuracy results were obtained with KNN. To categorize handwritten city names and recognize handwritten text, Daniyar Nurseitov et al. [15] created DL models. The first model uses deep CNN to classify handwritten cities, while the other three models use CNN layers, RNN layers, and the CTC decoding algorithm to recognize handwritten text. Wordbeamsearch yielded the best results, using a dictionary to fix the text under recognition at the end. In order to identify Kurdish handwritten characters, Rebin M. Ahmed et al. [16] used a DL algorithm. The model had a respectable accuracy rate, with 96% accuracy in testing and 97% accuracy in training.

Yasir Babiker Hamdan and Sathish [17] designed a framework for recognizing a variety of trendy characters in order to improve identification rate and accuracy. Better accuracy is successfully attained by the suggested framework by adding a fashionable character recognition process. When compared to current approaches, the suggested SVM-based HCR approach yields 94% accuracy and a decent identification rate. The DIGITNET framework, which uses DL network frameworks and involves DIGITNET-dect and DIGITNET-rec, was presented by Huseyin Kusetogullari et al. [18]. Historical handwritten document images are scanned using DIGITNET-dect to identify the digits and DIGITNET-rec to identify the digits that are detected. In DIGITNET-rec, three distinct CNN architectures are applied to the identified isolated digits, and the resulting classifications are then aggregated via a voting method to identify digit strings. Additionally, a variety of handwritten digit datasets are utilized to train the suggested model, which is subsequently verified using handwritten digit strings from the past. According to the experimental results, the suggested architecture that was trained using DIDA performed better than cutting-edge techniques. A machine learning model for handwritten alphabets was proposed by R. Popli et al. [19] for categorization and recognition. A small dataset is utilized to test the suggested methods, and the accuracy is determined using the KNN model. The model demonstrates an accuracy of 90% for identifying just three alphabets. The results are obtained by altering the value of k. The simulation's outcome showed that the best accuracy is obtained when an ensemble of KNN models is used. Thi Thi Zin et al. [20] proposed a handwritten character application that can be used as a teaching tool for children. In order to solve the segmentation issue, this research also suggested a character segmentation procedure. Character recognition can be accomplished with the CNN classifier. According to the simulation findings, the suggested system performed 95.6% on 1000 randomly chosen words and 98.7% on each character. Birjit Gope et al. [21] used the Mnist Database to offer a number of ML techniques for handwritten digit recognition. The proposed technique seeks to accurately and efficiently answer all the variables in terms of time complexity and accuracy. SVM obtained the best accuracy out of all the ML models.

Yevhen Chychkarov et al. [22] used MNIST datasets to develop a number of models, including the SVM Classifier, KNN Classifier, RF Classifier, MLP Classifier, and CNN, for the purpose of handwritten digit recognition. These algorithms showed nearly identical handwritten-digit identification accuracy after an easy setup, with variations of only +1%. When handwritten digit recognition was tested, CNN produced the most accurate results. The machine and DL approaches for handwritten digit identification from image input were developed by Meenal Jabde et al. [23]. In addition to using a pattern recognition strategy, four ML and DL algorithms are investigated. According to the investigation, the VGG 16 algorithm performs better in the DL-based technique than the ML method. The results show that the validation accuracy and loss are 0.2219 and 0.9602, respectively. A technique for offline handwritten Tai Le characters based on ensemble DL was proposed by Hai Guo et al. [24]. The handwritten Tai Le character dataset SDH2019.2 was initially created and pre-processed. After that, a stacking method was used to build an ensemble DCNN model. The findings showed that the deep neural network model based on stacking ensembles performs very well in Tai Le recognition. A character recognition method for the fundamental characters of old Devanagari manuscripts was created by Sonika Rani Narang et al. [25]. DL using the CNN model is utilized to recognize these characters. After 30 epochs, a maximum accuracy of 93.73% was attained using 75% train data and 25% test data. A DL and genetic algorithm-based model for Arabic HCR was proposed by Hossam Magdy Balaha et al. [26]. After a series of trial-and-error experiments, 14 distinct native CNN frameworks are suggested. The best-reported testing accuracy, based on experiments conducted on native CNN architectures, is 91.96%. The training parameters and hyperparameters in the recognition phase are recommended to be optimized using a genetic algorithm and transfer learning technique. The highest reported testing accuracy is 92.88%.

2.1 Research Gap

For training and testing, many of the experiments discussed rely on certain datasets (MNIST, NIST). The suggested models may not be as generalizable to other datasets or real-world situations, and their performance may differ depending on the handwriting style, language, or writing environment. A tiny dataset, or a dataset with few classes, is mentioned in certain studies. This may affect the suggested models' ability to be applied in real-world scenarios where data fluctuation is more pronounced. Hyperparameter tuning is a common step in DL models. Certain configurations that might not apply to other situations could be extremely dependent on optimal performance. Large-scale DL models in particular can consume a lot of resources, and there is little discussion of whether or not these models are practically feasible in settings with limited resources. There is still opportunity for performance improvement even though certain approaches—like MLPs—show better classification accuracy than others. In particular, they might have trouble with triangle-shaped numbers, high rotational changes in input samples, or big datasets, which would lead to a decrease in performance. Therefore, efforts are made continuously to improve the computational efficiency, accuracy, and robustness of HCR systems.

3 Materials and Methods

The availability of large-scale annotated data and contemporary architectures have been key factors in the success of DL-based models. This work methodically investigates many parameters in order to enhance handwritten identification for scanned off-line document images. This study suggested using a Deep CNN with Variable Filters and a Fusion based architecture for in-depth training. The main goal of this research is to generate an approach that combines feature fusion with DCNN based on variable filters to recognize handwritten characters. The accuracy of HCR is increased by using huge datasets. Dimensionality reduction and feature fusion are applied in order to achieve efficient recognition. DCNN-based image processing is used in this work for both image mapping and classification. Figure 3 depicts a detailed block schematic of the suggested methodology.

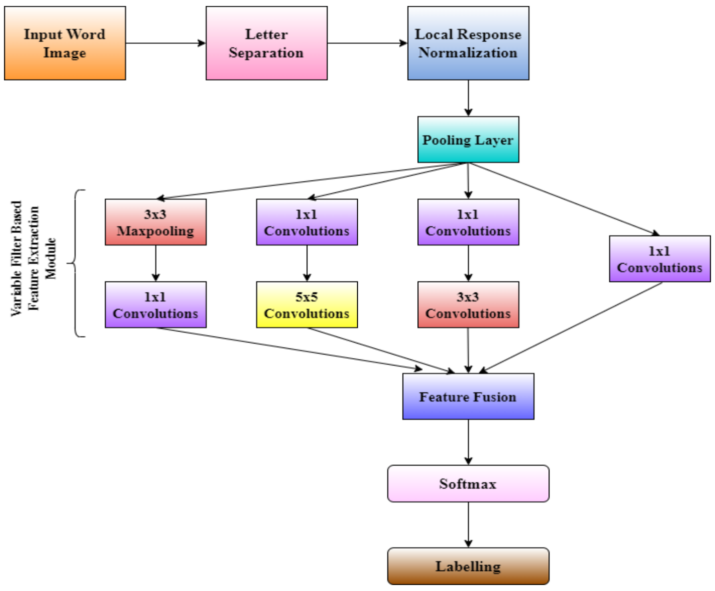


Fig.3. Detailed Block Schematics of Proposed Handwritten Character Recognition Model

The challenge of turning handwritten text contained in an image into machine-readable text is known as word recognition. The hybrid architecture used for HCR was a DNN-feature fusion system. This architecture is composed of a Local Response Normalization (LRN), a pooling layer, a series of convolutional blocks with varying filter sizes, feature fusion via concatenation, and a classifier to separate the labels. The optimal local construction is repeated spatially. In order to cluster the units into groups with high correlation, a layer-by-layer construction is used, where the correlation statistics of the preceding layer are examined. These clusters connect to the units in the preceding layer and form the units of the subsequent layer. To avoid patch alignment issues, the suggested architecture is restricted to filter sizes of 1x1, 3x3, and 5x5. In this architecture, the output filter banks of all those layers are combined using depth concatenation to produce a single output vector that is the input for the next step. Similar-height and -width inputs are concatenated along the channel dimension by a depth concatenation layer. The computational modules are piled on top of each other; hence, it is inevitable that their output correlation statistics will vary. According to the predicted decrease in spatial concentration, the ratio of 3x3 to 5x5 convolutions should rise as we go to higher layers. The quantity of output increases as one advances through the stages. The ideal sparse structure is covered by this design. The suggested architecture has the following peculiarities:

- An average pooling layer with a stride of three and a 5×5 filter size.
- A 1x1 convolution with 128 filters for dimension reduction and ReLu activation.
- A fully connected layer with 1024 units and ReLu activation.
- A dropout layer with 70% ratio of dropped outputs.
- A linear layer with softmax loss as the classifier.

3.1 Dataset Description

This research introduces a new dataset known as the "HandWritten_Character" dataset, which comprises image data. Emnist data has been utilized for the alphabets and digits. The data has been processed using various image processing techniques and converted into 32×32 pixel black and white images. This dataset was created to encompass special characters such as @, #, $, and &. The categories have been merged to prevent misclassification. The dataset includes all the English alphabets (both lowercase and uppercase), digits (0–9), and some special characters (@, #, $, &), totalling 39 categories. There are 26 categories for alphabets (combining lowercase and uppercase letters to create a single class for each character), 9 categories for digits (i.e., 1 to 9), and to prevent misclassification, digit 0 has been grouped with the character O category. In total, the dataset contains 834,036 images in the training folder and 22,524 images in the validation folder. The sample image from the dataset is visualized in Figure 4.

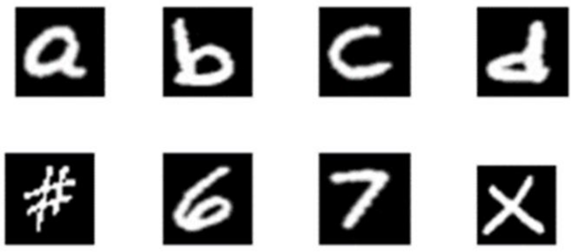


Fig.4. Handwritten Character Recognition Dataset

3.2 Letter Separation

A key phase in HCR is letter separation, where the objective is to recognize individual letters within a handwritten word or sequence. The continuous handwritten input is divided into discrete letter units during this process. Letter separation techniques include examining character spacing, looking for strokes that are vertical or horizontal, and using context to help pinpoint possible borders. Accurate letter separation is crucial to enhancing the performance of character recognition systems since each letter must be correctly isolated and identified in order for the handwritten text recognition process to function as accurately as possible.

3.3 Local Response Normalization (LRN)

A contrast enhancement method for feature maps in convNets is called local response normalization (LRN). A lateral inhibition mechanism that generates competition between neurons in a small local neighborhood in genuine biological neural networks serves as the inspiration for the LRN operation. On the feature map, the goal is to amplify "peaks" and reduce "flat" replies. As a result, LRN makes the brain more sensitive to desirable stimuli. Thus, if a strong peak response occurs, LRN fosters competition among the neurons in that neighborhood so that the strong response will inhibit the weak ones, amplifying the peak. However, if there is a uniformly strong response, then each of those strong reactions will essentially cancel each other out, making the neighborhood as a whole less active. Biological vision served as the inspiration for the introduction of this sort of lateral inhibition, which is especially prevalent in CNNs. The intention is to stimulate competition between nearby neurons, thus amplifying the activity of certain neurons and inhibiting that of other neighbors.

Given an input feature map x_i at spatial position i and channel c, normalized output y_i^c is computed as:

$$y_i^c = \frac{x_i^c}{\left(\left(k+\alpha \sum_{j=\max(0,c-n/2)}^{\min(N-1,c+n/2)} (x_j^i)^2\right)^{\beta}\right.} \qquad (1)$$

Here, x_i^c is the activity of the neuron at position i in channel c, N is the total number of channels, n is the size of the local neighbourhood (usually an odd number), k, α, β are the hyperparameters that control the normalization. The block schematics of the LRN is visualized is illustrated in Figure 5.

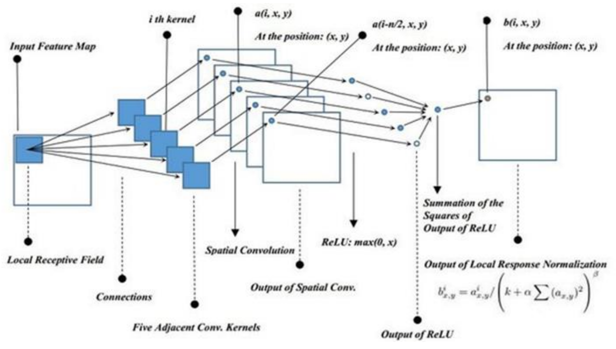


Fig.5. Block Diagram of Local Response Normalization

3.4 Convolutional Neural Network (CNN)

CNNs, a type of hierarchical feed-forward network, use unique operators known as convolutions to extract salient elements from images. These convolution operators enable CNN to process visual input immediately, with little need for pre-processing. Furthermore, dimension reduction and classification are carried out via a series of layers that receive the convoluted images. The different layers in a CNN are visualized in Figure 6.

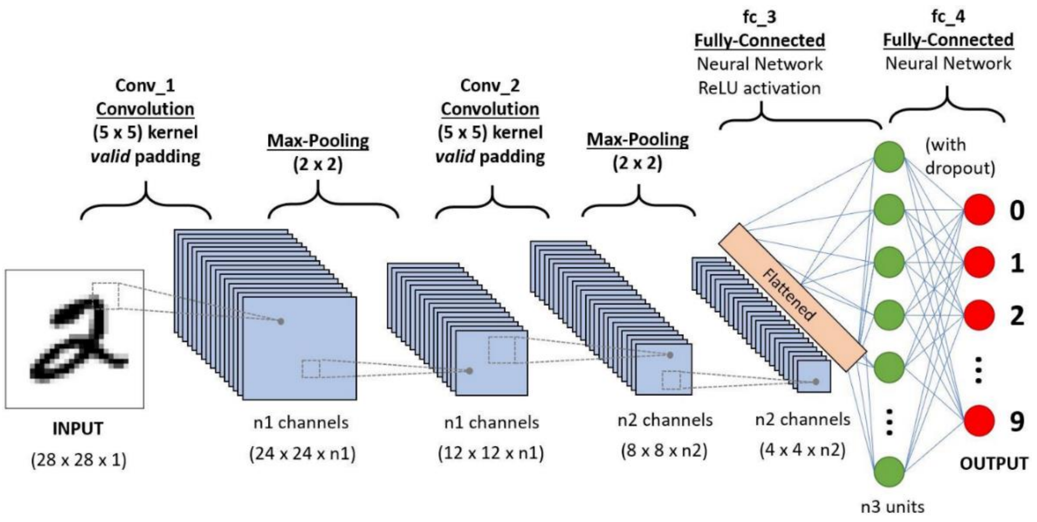


Fig.6. Basic Architecture of Convolutional Neural Network

In order to identify visual elements like lines, edges, and various colour shades, the first layer of the CNN processes input in the form of raw pixel values. This layer maintains the correlations between pixels through repeated learning employing small squares of values, forming patterns known as kernels or filters [27]. Equation 2 describes how these kernels are cross-multiplied with the input pixel values.

$$\mathrm{C} = \mathrm{I} \otimes \mathrm{F} \qquad (2)$$

where F stands for the feature matrix, also known as the kernel, and I stand for the input matrix containing the pixel values. The output of the convolution process is represented by C. Image convolution using various kernels can be used to carry out tasks including edge detection, sharpening, and blurring. The feature map is passed into the ReLu function at the end of this layer in order to increase non-linearities. A CNN's ability to use the pooling operation to preserve significant patterns in images even at high resolutions is one of its distinctive properties. The pooling operation is used to achieve dimensionality reduction [28]. CNNs frequently employ max pooling, a down sampling technique, to minimize the spatial dimensions of feature maps while preserving the most crucial data. The maximum value is kept for each section of the input feature map where the pooling operation is applied independently. This aids in identifying the salient traits and eliminating irrelevant data. It can be mathematically represented as:

$$Max\ Pooling\ (x)_{(i,j)} = \max(x_{2i,2j}, x_{2i,2j+1}, x_{2i+1,2j}, x_{2i+1,2j+1}) \quad (3)$$

Here, $x_{(i,j)}$ represents the element at the i^{th} row j^{th} column of the input feature map x. The visualization of max pooling is visualized in Figure 7.

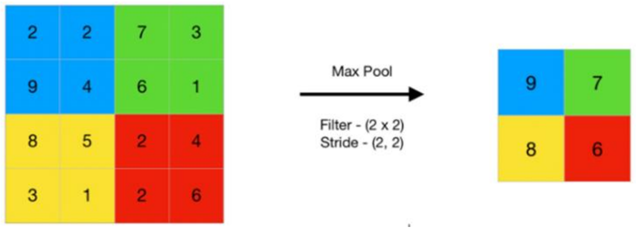


Fig.7. Max Pooling Operation

CNNs frequently employ average pooling, a down sampling technique, to minimize the spatial dimensions of feature maps while sustaining essential information. The procedure involves taking the average value of each zone after splitting the input feature map into non-overlapping parts, usually squares. A new, down-sampled feature map with fewer spatial dimensions is created from the resultant data. Let X be the input feature map, $n \times n$ be the size of the pooling window, and Y be the resulting output feature map. For each region in X defined by the pooling window, the average pooling operation is represented by:

$$Y_{(i,j)} = \frac{1}{n \times n} \sum_{p=0}^{n-1} \sum_{q=0}^{n-1} X_{i+p,j+q} \quad (4)$$

Here, $Y_{(i,j)}$, signifies value at position (i,j) in the down sampled feature map, and $X_{i+p,j+q}$ signifies the pixel value at position $(i+p, j+q)$ in the input feature map. The visualization of average pooling is illustrated in Figure 8.

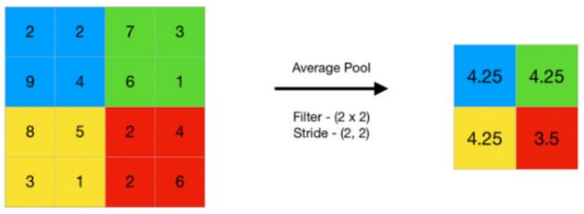


Fig.8. Average Pooling Operation

The fully connected layer eventually flattens the combined feature map into a single-dimensional vector. Since a classifier uses a single-dimensional vector to categorize the images, it is sometimes referred to as the classification layer.

3.5 Feature Fusion

The process of merging numerous sets of features from many sources or layers into a single, cohesive representation is called feature fusion, sometimes referred to as feature concatenation or feature combination. This procedure is frequently used to gather supplementary data and enhance a model's overall performance in DL architectures such as CNNs or RNNs. A neural network may experience feature fusion at the input level, intermediate layers, or output layer, among other phases. Features that have been retrieved from various network segments may be merged into intermediary layers. This is frequently used to combine low-level spatial features with high-level semantic information. More intricate hierarchical representations can be learned by the model through feature fusion at intermediate layers. A single feature vector is created by fusing the output features from the 1x1, 3x3, and 5x5 images. By concatenating the output filter banks of each level into a single output vector, which functions as the input for the next stage, the architecture that is being presented appears to be a synthesis of all those levels. The visualization of feature fusion operation is shown in Figure 9.

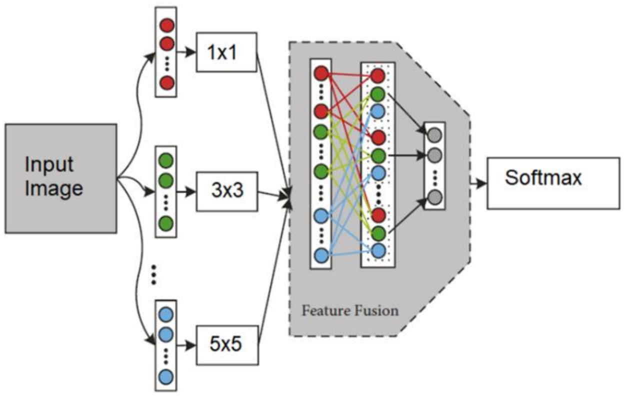


Fig.9. Feature Fusion Operation

3.6 Softmax Classifier and Labelling

The final step of the classification process in HCR is greatly dependent on the Softmax classifier. A vector of raw scores, or logits, is converted into probabilities using the mathematical function softmax. Each score indicates the chance that the input belongs to a specific class. A probability distribution over the potential character classes is produced by the softmax classifier in the context of HCR using the output from earlier neural network stages, such as convolutional and fully connected layers. As a result, the model can decide with confidence which character the handwritten input most likely represents. It is simple to interpret the output as a probability distribution and choose the most likely class as the final prediction because the softmax function makes sure that the projected probabilities add up to 1.

In HCR, labelling entails giving the training set of data ground truth labels, which are necessary for the model to properly learn and generalize. In the training set, every handwritten character image has a unique label that reveals the character's actual identity. During the training phase, the model uses these labels as reference points to minimize the discrepancy between its predictions and the ground truth labels. With the help of the labelled dataset, the model is capable of identify and classify new, unseen handwritten characters with accuracy as it learns the patterns and attributes associated with each character class. The robustness of the softmax classifier in producing precise probability distributions for character classification, along with the calibre and representativeness of the labelled training data, are major factors influencing the model's efficacy.

The detailed algorithm of the proposed handwritten character recognition model is explained below.

```
Input: Word image containing handwritten characters
Output: Recognized characters and their corresponding labels.

Begin
    • Letter Separation: Perform letter segmentation to isolate individual characters.
    • Local Response Normalization (LRN): For each character image:
        ❖ Apply local response normalization to enhance local contrast.
        ❖ Local response normalization formula for a pixel at position (i, j) in a feature map:
          $LRN\ (i,j) = \frac{x(i,j)}{(k+\alpha(sum(x(i,j))^2))^\beta}$
    • Pooling Layer: Apply pooling to reduce spatial dimensions.
        ❖ Max pooling or average pooling can be used.
        ❖ Pooling operation for a pixel at position (i, j): Pooling(i, j) = max or average over a local neighborhood
          centered at (i, j)
    • Variable Filters-Based Feature Extraction:
        ❖ Initialize variable filters with learnable parameters- - Let W be the set of filters.
        ❖ Convolutional operation for a feature map F-
              F (i, j) = activation_function (sum (W * input_patch) + bias)
        ❖ Apply activation function (e.g., ReLU) after convolution.
    • Feature Fusion:
        ❖ Extract features from different convolutional layers- Let F1, F2, ..., Fn be feature maps from different
          layers.
        ❖ Concatenate or sum features for fusion-
          F_fused = concatenate (F1, F2, ..., Fn) or F_fused = F1 + F2 + ... + Fn
    • Softmax Classifier:
        ❖ Flatten the fused features for input to the softmax classifier- Let flattened_features be the flattened feature
          vector.
        ❖ Apply softmax function for classification.
        ❖ Choose the class with the highest probability as the predicted label.
    • Labelling:
        ❖ Assign the recognized label to the corresponding character in the word image.
End
```

4 Results and Discussion

4.1 Hardware and Software Setup

There were many difficulties with the implementation, including the need to modify the development environment to meet the specifications. The experiments and simulations were conducted using Python language. Since the image data was supplied in the form of .jpg files. The process of downloading it needed to be programmed in a Python environment. To provide access to the images during the model-building phase, every single image was transferred to Google Drive. In addition, a workstation with a top-of-the-line setup that included an Intel i7 processor, 32GB of RAM, and an 8GB Nvidia graphics processing unit was used for image processing and exploration.

The neural network model uses the Adam optimizer with a 0.001 learning rate and 10,211,534 trainable parameters. The training process is guided by the categorical cross-entropy loss function. During training, the model processes input data in batches of 64 samples per iteration. The training is carried out over 50 epochs, signifying the number of times the entire training dataset is processed by the model. These hyperparameter choices, such as the optimizer, learning rate, loss function, batch size, and number of epochs, collectively define the configuration for training the neural network model, aiming to optimize its performance on the given HCR task. The model configuration of the suggested approach is tabulated in Table 1.

Table 1. Model Configuration

Model Parameters	**Proposed Model (CNN-Bi GRU)**
Trainable Parameters	10,211,534
Optimizer	Adam
Learning Rate	0.001
Loss Function	Categorical Cross Entropy
Batch Size	64
Number of Epochs	50

4.2 Performance Evaluation

The accuracy plot shows how the neural network model performed during the HCR training process. The accuracy rises steadily after 50 epochs, demonstrating the model's capacity to identify characters in the training dataset accurately. The figure illustrates how accuracy converges and stabilizes with time, providing a visual representation of the learning process. A graphical representation is crucial in order to assess the training dynamics of the model and spot possible problems like overfitting or underfitting. As the model iteratively processes the training data, the accuracy plot indicates that the model is effectively learning and improving its predicting capabilities. The accuracy plot of the suggested model is visualized in Figure 10.

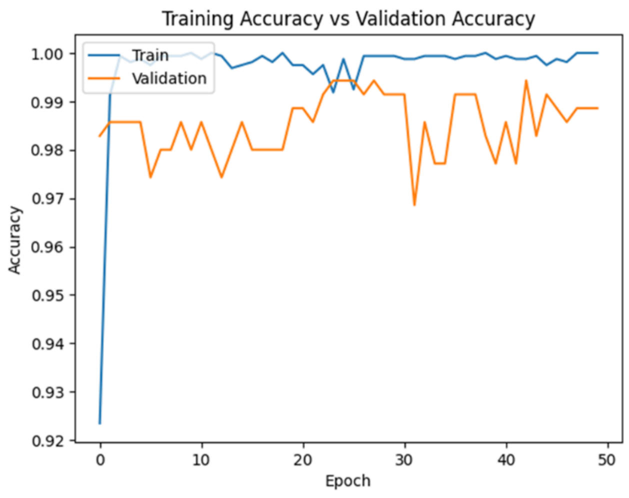


Fig.10. Accuracy plot of Proposed Model

The loss plot shows how the categorical cross-entropy loss function changed during the neural network model's training process. As the model gains experience in accurately representing the relationships found in the training data, the loss progressively drops from the beginning value. The plot illustrates how effectively the model is adjusting to the particular HCR challenge and offers insights into the convergence and stability of the training process. A declining trend in the loss plot indicates that the model's prediction accuracy is getting better, whereas oscillations or sharp spikes can point to problems or changes in the training dynamics. The general form of the loss plot is an important diagnostic tool that evaluates the efficacy of the selected hyperparameters as well as the training process. The loss plot of the suggested model is visualized in Figure 11.

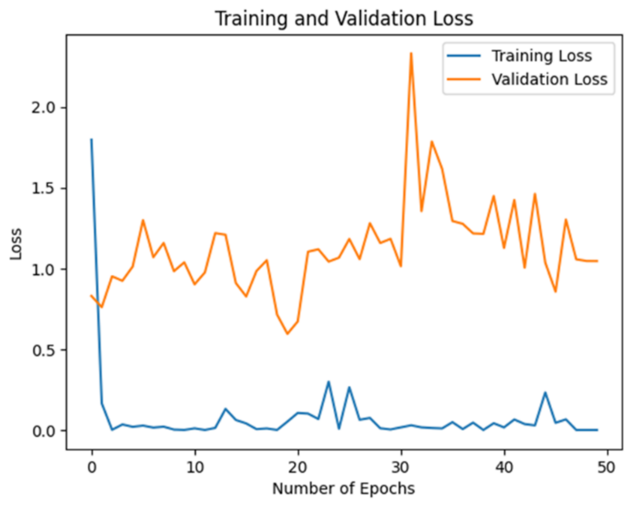


Fig.11. Loss plot of Proposed Model

The capacity of the model to accurately recognize and categorize characters in a dataset is referred to as its accuracy. The fraction of accurately predicted characters relative to all the characters in the dataset is typically given as a percentage. A model that is better at identifying handwritten characters is indicated by a higher accuracy value. In the context of character recognition tasks, the accuracy metric offers a quantifiable assessment of the model's performance, demonstrating its capacity to produce accurate predictions and generalize well to new data. The accuracy can be mathematically expressed as:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \quad (5)$$

The Proposed Hybrid Feature Fusion model provided an average accuracy of 98.86%. The images from validation folder are used for the prediction of characters in the hand-written word. The sort contours function is used to get the correct order of individual characters for correct output extraction. Table 2 tabulates the performance comparison of the suggested HCR model with other two approaches (CNN+Bi-LSTM and CNN+ Bi- GRU).

Table 2. Performance Comparison

Proposed Models	**Accuracy (%)**
CNN+Bi-LSTM	91.35
CNN+Bi-GRU	96.72
Deep CNN with variable filters and feature fusion model	98.86

The provided accuracy results showcase the performance of different proposed approaches for HCR. The CNN+ Bi-LSTM model attained an accuracy of 91.35%, indicating a strong capability in recognizing handwritten characters. The CNN+ Bi-GRU model demonstrated an even higher accuracy at 96.72%, suggesting improved performance over the CNN+ Bi-LSTM model. Notably, the Deep CNN with Variable Filters and Feature Fusion Model exhibited the highest accuracy among the proposed models, reaching an impressive 98.86%. This signifies the effectiveness of incorporating DCNN with variable filters and feature fusion techniques in enhancing the accuracy of HCR. The outcomes suggest that the latter model captures more intricate patterns and dependencies in the handwritten characters, leading to superior classification accuracy compared to the other proposed architectures. Overall, the outcomes underscore the importance of model architecture choices in optimizing performance for HCR tasks. The performance comparison is graphically illustrated in Figure 12. The performance comparison of the suggested HCR model with existing methods are tabulated in Table 3.

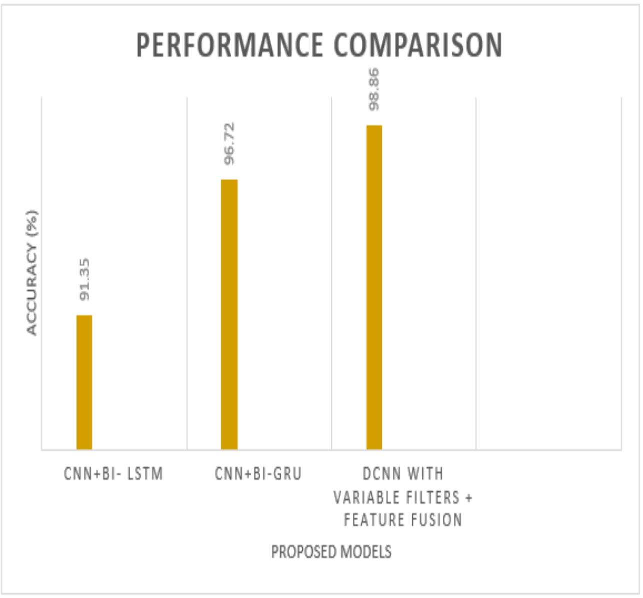


Fig.12. Performance Comparison of Proposed Models

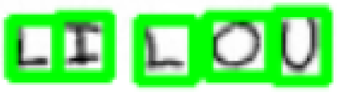
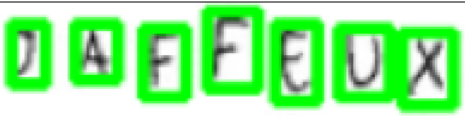
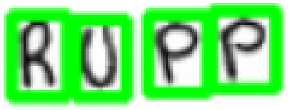
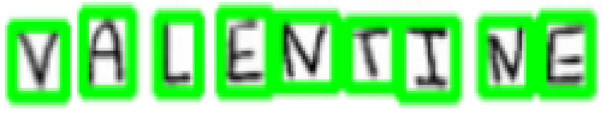
Table 3. Performance Comparison with Existing HCR Approaches

Authors and Year	**Methodology**	**Accuracy (%)**
S M Shamim *et al.* (2018) [29]	Support Vector Machine, J48, Naive Bayes, Random Forest, Multilayer Perceptron, Bayes Net	90.37
Yasir Babiker Hamdan and Sathish (2021) [17]	Statistical Support Vector Machine	91
Hossam Magdy Balaha *et al.* (2021) [26]	Deep Learning and Genetic Algorithm	92.88
I Khandokar *et al.* (2021) [13]	Convolutional Neural Network	92.91
Rebin M. Ahmed *et al.* (2022) [16]	Convolutional Neural Network	96
Huseyin Kusetogullari *et al.* (2021) [18]	Convolutional Neural Network and YOLO	97.12
Mayur Bhargab Bora	Convolutional Neural Network and Error	97.71

et al. (2020) [30]	Correcting Output Code Classifier	
Proposed Methodologies:		
1. CNN+ Bi- LSTM		
2. CNN+ Bi-GRU		**91.35**
3. Deep CNN with Variable filters And Feature Fusion Model		**96.72** **98.86**

Some of the recognition outputs using the proposed HCR model is tabulated in Table 4.

Table 4. Recognition Outputs

Input Image	Resulting Text
	BENES
	LILOU
	JAFFEUX
	RUPP
	VALENTINE

5 Conclusion

HCR holds immense significance in the realms of technology, communication, and accessibility. As we continue to transition into a digital age, the ability to accurately and efficiently convert handwritten text into digital format becomes increasingly crucial. HCR plays a pivotal role in diverse applications, ranging from digitizing historical manuscripts and preserving cultural heritage to enhancing data input methods in various industries. HCR is a technology that involves the automatic identification and interpretation of characters, symbols, or words written by hand. Handwritten text has to be transformed into machine-encoded text in order for HCR to be readable and useful in digital formats. This is a subset of OCR, which is the field that generally deals with handwritten and printed text recognition. This paper presented an effective HCR model using DCNN with variable filter and feature fusion. A handcrafted HCR dataset can be used for model evaluation. Finally, a comparison of the suggested model's performance with the CNN+ Bi-LSTM and CNN+ Bi-GRU models is conducted. The simulation results shown that the Deep CNN with Variable Filters and Feature Fusion Model emerges as the most successful, achieving an impressive accuracy of 98.86%. The CNN+ Bi-LSTM and CNN+ Bi-GRU models showcase respectable accuracies at 91.35% and 96.72%, respectively, emphasizing the effectiveness of combining convolutional and sequential modelling techniques. The results suggest that the Deep CNN with Variable Filters and Feature Fusion Model stands out as a highly effective architecture for HCR, surpassing the performance of the other suggested approaches. The findings underline the significance of sophisticated model architectures in achieving optimal accuracy in the context of HCR tasks.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Srihari, S. N., Cha, S. H., Arora, H., & Lee, S. (2002). Individuality of handwriting. Journal of forensic sciences, 47(4), 856-872.

[2] Pal, A., & Singh, D. (2010). Handwritten English

character recognition using neural network. International Journal of Computer Science & Communication, 1(2), 141-144.

[3] OAM, P. W., & Crew, G. (2018). The future of handwriting. TEXT, 22(Special 52), 1 10.

[4] Pal, U., Jayadevan, R., & Sharma, N. (2012). Handwriting recognition in indian regional scripts: a survey of offline techniques. ACM Transactions on Asian Language Information Processing (TALIP), 11(1), 1-35.

[5] Chaudhuri, A., Mandaviya, K., Badelia, P., K Ghosh, S., Chaudhuri, A., Mandaviya, K., ... & Ghosh, S. K. (2017). Optical character recognition systems (pp. 9-41). Springer International Publishing.

[6] Aparna, K. H., Subramanian, V., Kasirajan, M., Prakash, G. V., Chakravarthy, V. S., & Madhvanath, S. (2004, October). Online handwriting recognition for Tamil. In Ninth international workshop on frontiers in handwriting recognition (pp. 438-443). IEEE.

[7] Chatterji, B. N. (1986). Feature extraction methods for character recognition. IETE Technical Review, 3(1), 9-22.

[8] Graves, A., & Schmidhuber, J. (2008). Offline handwriting recognition with multidimensional recurrent neural networks. Advances in neural information processing systems, 21.

[9] Gupta, A., Srivastava, M., & Mahanta, C. (2011, December). Offline handwritten character recognition using neural network. In 2011 IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE) (pp. 102-107). IEEE.

[10] Maitra, D. S., Bhattacharya, U., & Parui, S. K. (2015, August). CNN based common approach to handwritten character recognition of multiple scripts. In 2015 13th International Conference on Document Analysis and Recognition (ICDAR) (pp. 1021-1025). IEEE.

[11] Nishide, S., Okuno, H. G., Ogata, T., & Tani, J. (2011, October). Handwriting prediction-based character recognition using recurrent neural network. In 2011 IEEE International Conference on Systems, Man, and Cybernetics (pp. 2549-2554). IEEE.

[12] Pashine, S., Dixit, R., & Kushwah, R. (2021). Handwritten digit recognition using machine and deep learning algorithms. arXiv preprint arXiv:2106.12614.

[13] Khandokar, I., Hasan, M., Ernawan, F., Islam, S., & Kabir, M. N. (2021, June). Handwritten character recognition using convolutional neural network. In Journal of Physics: Conference Series (Vol. 1918, No. 4, p. 042152). IOP Publishing.

[14] KARAKAYA, R., & KAZAN, S. (2021). Handwritten digit recognition using machine learning. Sakarya university journal of science, 25(1), 65-71.

[15] Nurseitov, D., Bostanbekov, K., Kanatov, M., Alimova, A., Abdallah, A., & Abdimanap, G. (2021). Classification of handwritten names of cities and handwritten text recognition using various deep learning models. arXiv preprint arXiv:2102.04816.

[16] Ahmed, R. M., Rashid, T. A., Fattah, P., Alsadoon, A., Bacanin, N., Mirjalili, S., ... & Chhabra, A. (2022). Kurdish Handwritten character recognition using deep learning techniques. Gene Expression Patterns, 46, 119278.

[17] Hamdan, Y. B., & Sathesh, A. (2021). Construction of statistical SVM based recognition model for handwritten character recognition. Journal of Information Technology and Digital World, 3(2), 92-107.

[18] Kusetogullari, H., Yavariabdi, A., Hall, J., & Lavesson, N. (2021). DIGITNET: A deep handwritten digit detection and recognition method using a new historical handwritten digit dataset. Big Data Research, 23, 100182.

[19] Popli, R., Kansal, I., Garg, A., Goyal, N., & Garg, K. (2021). Classification and recognition of online hand-written alphabets using machine learning methods. In IOP Conference Series: Materials Science and Engineering (Vol. 1022, No. 1, p. 012111). IOP Publishing.

[20] Zin, T. T., Thant, S., Pwint, M. Z., & Ogino, T. (2021). Handwritten character recognition on android for basic education using convolutional neural network. Electronics, 10(8), 904.

[21] Gope, B., Pande, S., Karale, N., Dharmale, S., & Umekar, P. (2021). Handwritten digits identification using MNIST database via machine learning models. In IOP conference series: materials science and engineering (Vol. 1022, No. 1, p. 012108). IOP Publishing.

[22] Chychkarov, Y., Serhiienko, A., Syrmamiikh, I., & Kargin, A. (2021). Handwritten Digits Recognition Using SVM, KNN, RF and Deep Learning Neural Networks. CMIS, 2864, 496-509.

[23] Jabde, M., Patil, C., Mali, S., & Vibhute, A. (2022, August). Comparative Study of Machine Learning and Deep Learning Classifiers on Handwritten Numeral Recognition. In International Symposium on Intelligent Informatics (pp. 123-137). Singapore:

Springer Nature Singapore.

[24] Guo, H., Liu, Y., Yang, D., & Zhao, J. (2022). Offline handwritten Tai Le character recognition using ensemble deep learning. The Visual Computer, 38(11), 3897-3910.

[25] Narang, S. R., Kumar, M., & Jindal, M. K. (2021). DeepNetDevanagari: a deep learning model for Devanagari ancient character recognition. Multimedia Tools and Applications, 80, 20671-20686.

[26] Balaha, H. M., Ali, H. A., Youssef, E. K., Elsayed, A. E., Samak, R. A., Abdelhaleem, M. S., ... & Mohammed, M. M. (2021). Recognizing Arabic handwritten characters using deep learning and genetic algorithms. Multimedia Tools and Applications, 80, 32473-32509.

[27] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of big Data, 8, 1-74.

[28] Scherer, Dominik, Andreas Muller, and Sven Behnke, "Evaluation of pooling ¨ operations in convolutional architectures for object recognition," International Conference on Artificial Neural Networks, Springer, Berlin, Heidelberg, pp.92- 101, 2010.

[29] Shamim, S. M., Miah, M. B. A., Sarker, A., Rana, M., & Al Jobair, A. (2018). Handwritten digit recognition using machine learning algorithms. Global Journal Of Computer Science And Technology, 18(1), 17-23.

[30] Bora, M. B., Daimary, D., Amitab, K., & Kandar, D. (2020). Handwritten character recognition from images using CNN-ECOC. Procedia Computer Science, 167, 2403-2409.