SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V11I1P101

Volume 11 Issue 1, 1-14, January 2024

Original Article

A Novel Approach: CNN-RNN and Bi-GRU for Handwritten Character Recognition

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

[1,2]*Department of Digital Sciences, Karunya Institute of Technology and Science, Tamil Nadu, India.*

[1]*Corresponding Author : sunithasnair24@gmail.com*

Received: 14 October 2023 Revised: 24 November 2023 Accepted: 15 December 2023 Published: 13 January 2024

Abstract - *Handwritten Character Recognition (HCR) plays a crucial role in converting handwritten content into machine-readable format with applications spanning various sectors such as historical document digitization, postal envelope address reading, form processing, and assisting individuals with disabilities. Recognizing handwritten characters is inherently challenging due to the diversity in writing styles and shapes and the presence of noise in the data. Recent advances in deep learning, particularly the utilization of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have greatly enhanced the accuracy and performance of HCR systems. The integration of a CNN-RNN model with Bidirectional Gated Recurrent Units (Bi-GRU) has shown great promise, achieving an impressive accuracy rate of 96.72%. The CNN component excels at capturing spatial features and character structures, while the RNN with Bi-GRU layers effectively models sequential dependencies in handwritten text. As technology continues to advance and more data becomes available, the future holds the promise of even more refined and powerful HCR models. This hybrid approach has the potential to automate processes, enhance data processing, and improve user experiences in a wide range of industries.*

Keywords - *Character shapes, Deep Learning, Handwritten data, Optical Character Recognition, Sequential dependencies.*

1. Introduction

Handwriting, a timeless form of communication, has evolved over the ages to align with cultural and technological progress. Its core attributes can be distilled into three key elements: the deliberate creation of graphical symbols on a medium, the intention to convey a message, and the achievement of this aim through the established association of these symbols with language [1]. The origins of writing can be traced back to ancient times, serving as a fundamental means for individuals to express their emotions and enhance communication.

This advancement in written communication played a pivotal role in shaping culture and civilization, offering the invaluable ability to document history, cultural aspects, literature, legal systems, scientific knowledge, mathematical concepts, and more. Writing is essentially a system of standardized symbols, following specific rules to represent ideas [2].

As cultures and civilizations advanced over time, the necessity for enhanced communication gave rise to the development of languages, scripts, and alphabets. The act of writing, or penmanship, has evolved into a distinctive and personalized skill, varying from one individual to another. It is influenced by a multitude of factors, such as one's mental state, mood, the writing medium, and the environment. Despite the digital age, the enduring relevance of handwriting can be attributed to the convenience of using pen and paper, which remains preferable in various day-to-day situations compared to keyboards.

There are worries regarding the future of traditional handwriting because of the undeniably revolutionary changes in communication and information recording brought about by the widespread usage of digital computers. However, there are numerous situations where the simplicity of a pen and paper or a compact notepad remains unmatched in terms of convenience when compared to a keyboard [3].

The true challenge for computer systems lies in their ability to decipher and work with unconstrained handwritten text, accommodating various writing styles, languages, and even user-defined alphabets. They must be capable of understanding handwritten messages from any individual. Due to the continuous advancement of computer technology and the affordability of high-capacity storage devices, electronic document storage has become a common practice.

In both handwritten and printed forms, documents often incorporate a mixture of various scripts, graphics, and images. Even in the face of this digital revolution, physical documents continue to play a significant role in various forms of communication. They remain indispensable tools for many individuals. The enduring preference for paper as a reliable and secure medium for document management ensures that the demand for physical records will persist for years to come [4].

Therefore, there is a significant demand for software solutions that can automate the extraction, analysis, and storage of information from physical documents, making later retrieval more streamlined. These documents may contain multiple languages, with each script defined by a unique set of characters or letters, each having distinct basic shapes, as visualized in Figure 1. The creation of software capable of efficiently managing the diversity of scripts and characters in such documents is crucial to addressing the evolving requirements of an increasingly digital world.

The automatic recognition of handwritten information in documents holds significant practical and commercial importance across various sectors [5]. Organizations dealing with a high volume of such documents on a daily basis rely on automated reading systems, which have become indispensable. Even if these systems can accurately recognize only a portion of the documents, they offer substantial time and labour savings.

The increasing prevalence of devices such as smartphones and digital computers has further heightened the demand for Handwritten Character Recognition (HCR). HCR involves scanning input from images, documents, and real-time devices like tablets, digitizers, and tabloids and converting them into digital text. Handwritten Optical Character Recognition (OCR) represents a computer's ability to receive, decipher, and effectively analyze handwritten input through automated processes [6]. A typical OCR system comprises multiple sequential steps, as depicted in Figure 2. The evolution of this technology enables the efficient conversion of handwritten data into digital formats, streamlining operations and enhancing information accessibility across various sectors.

HCR can be divided into two distinct categories based on the method of data acquisition, as shown in Figure 3. These two groups are defined by the way handwritten characters are input into the recognition system. Online HCR involves the identification of digitized handwriting using various methods. Typically, a specialized pen is used in conjunction with an electronic surface to capture and store handwritten content in digital form.

As the pen moves across the surface, it records the 2D coordinates of successive points over time and preserves their sequential order. It is worth noting that online data acquisition tends to be less noisy and requires minimal preprocessing [7]. The algorithms used for tasks such as skew correction and normalization must be efficient and speedy. Furthermore, the use of pen trajectory data simplifies and expedites the feature extraction processes, including stroke detection, determination of stroke orientations, and identification of corners, loops, cusps, and other relevant features.

Fig. 1 Sample handwritten features

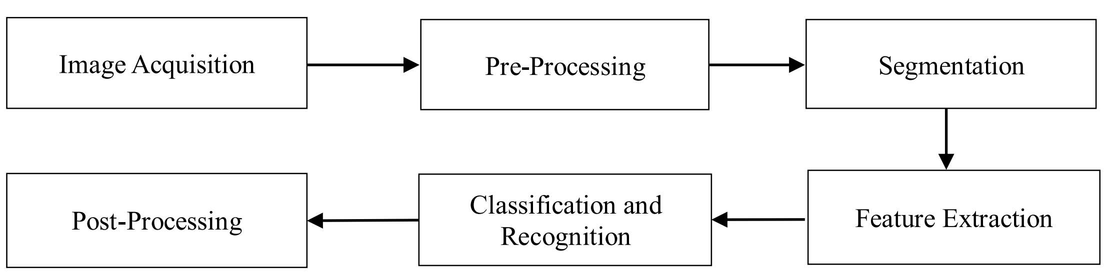


Fig. 2 General schematics of character recognition system

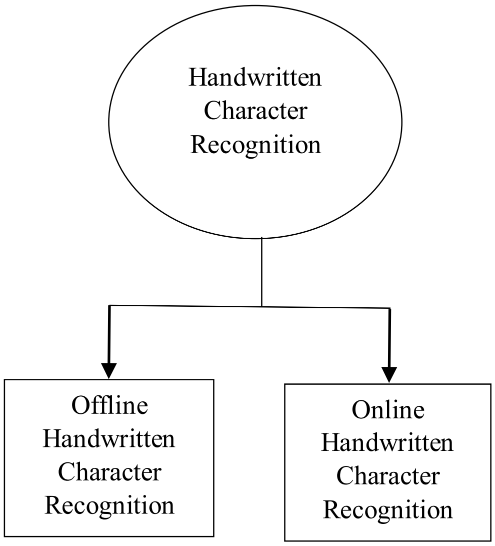


Fig. 3 Different categories of Handwritten Character Recognition

Offline handwriting recognition involves the identification of words scanned from a physical source, such as a paper sheet, and saved in a digital grayscale format. After the storage phase, it is customary to engage in additional processing to enhance recognition quality. The input image often exhibits noise due to various contributing factors, including the writer's style, paper quality, choice of pen and ink, and the scanner's characteristics, all of which significantly impact the recognition process [8]. Notably, temporal information is typically lacking, making it challenging to extract strokes from the image in a coherent sequence. It is widely recognized that online handwritten text recognition methods tend to deliver superior results compared to their offline counterparts. The superiority of online character recognition can be attributed to its ability to capture more detailed information.

Moreover, offline systems can also leverage features such as positional, geometrical, and statistical characteristics due to the availability of the complete image. The primary distinction between online and offline character recognition lies in their handling of contextual information. In online character recognition, real-time context plays a pivotal role, whereas offline character recognition processes data devoid of dynamic contextual cues. This disparity necessitates distinct processing architectures and methodologies for each approach.

People's handwriting varies for many reasons. Furthermore, there exists a group of characters that bear a striking resemblance to each other, with only minute differences setting them apart. Even for humans, distinguishing between these characters proves to be a challenging task. These variations affect recognition accuracy. Some of the challenges faced by the researchers in automating an HCR system are given below. Due to the unlimited variety of character shapes, the intrinsic similarity between characters, and the existence of distorted and unreadable characters, HCR is a challenging task. Challenges are further compounded with broken, overlapping, and touching characters, as well as the existence of old letters and compound characters within documents.

Additionally, the coexistence of different languages and digits in a single dataset, coupled with variations in document quality and the percentage of noise content, pose significant hurdles for accurate recognition. To address these challenges, recognition algorithms often employ complex methods, resulting in high computational complexity and increased memory requirements due to the large number of character classes. Consequently, the demand for extensive training on large databases is necessary, but it comes at the cost of a high computational burden and lengthy training times.

Although some methods, like MLPs, exhibit superior classification accuracy compared to other techniques, there remains room for performance improvement. Specifically, they may struggle when dealing with numerals forming a triangle shape, intense rotational variations in input samples, or large datasets, causing a decline in method performance. As such, efforts are continually made to enhance the accuracy, robustness, and computational efficiency of HCR systems.

Deep Learning (DL) [9] has revolutionized the field of HCR by demonstrating its remarkable ability to decipher intricate patterns autonomously and representations from raw data. When applied to HCR, DL models exhibit outstanding performance in efficiently capturing both spatial and temporal dependencies inherent in handwritten characters. These neural networks have the inherent capacity to independently acquire essential features and construct hierarchical representations from input data, significantly reducing the need for labour-intensive manual feature engineering, which has been a longstanding challenge in traditional character recognition methods.

As a result, the adoption of DL techniques has substantially improved the accuracy and robustness of HCR systems, making them highly adaptable to various handwriting styles, languages, and fonts. Furthermore, DL models demonstrate an impressive ability to seamlessly handle large datasets, opening up a wide range of applications for HCR. These applications range from automating document transcription to OCR systems dedicated to the digitization of both printed and handwritten text. The scalability of these models, combined with their potential for transfer learning and fine-tuning, positions DL as an essential tool for creating more accurate, versatile, and efficient solutions in the field of HCR. Beyond just improving recognition accuracy, the significance of DL extends to streamlining development processes and enhancing the adaptability of recognition systems. As a result, these systems gain increasing importance across various fields, including document management, text analysis, and language processing [10]. The primary contribution of this research entails:

- Development of a framework for the recognition of handwritten characters by combining Bi-GRU based RNN and CNN.
- Utilization of large datasets to improve the accuracy of HCR.
- Employment of Deep Neural Network (DNN) based image processing for image classification and mapping.
- Performance comparison with state-of-the-art methods

The remainder of the paper follows this structure: Section two provides a literature review, highlighting areas necessitating further research. Section three elucidates the methodology in detail. Section four delves into a comprehensive discussion of the outcomes resulting from the proposed approach. Lastly, in Section five, the paper concludes by summarizing the findings.

2. Related Works

Yasir Babiker Hamdan and Sathish [11] introduced a novel framework designed to enhance the recognition of diverse and stylish characters, ultimately leading to improved accuracy rates. The innovative approach incorporates a specialized procedure for identifying stylish characters, catering to handwriting styles characterized by distorted strokes and varying thicknesses in italic characters. To attain higher accuracy levels, the input image undergoes meticulous preprocessing and feature extraction. Simulation outcomes underscore the effectiveness of the proposed SVM-based HCR approach, which achieves an impressive 94% accuracy, outperforming existing methodologies in terms of recognition rates.

Khandokar et al. [12] developed a CNN with the primary objective of assessing its proficiency in character recognition using the NIST dataset, aiming for heightened accuracy. The simulation findings demonstrated a progressive enhancement in accuracy as the number of training images increased. Specifically, starting with 200 training images, the accuracy stood at 65.32% and steadily improved. It culminated at an impressive 92.91% when employing 1000 training images.

Saleh Albahli et al. [13] introduced an innovative technique for automating the recognition and categorization of handwritten digits. The proposed method harnessed the power of DL, combining the Faster-RCNN technique with the feature extraction capabilities of the DenseNet-41 framework. The methodology utilized Faster-RCNN for the precise classification of numerals, while DenseNet-41 was employed to compute deep features and facilitate digit detection. The outcome of this approach was highly influential in accurately localizing digits within input images and assigning them to one of the ten classes, each representing integer values ranging from 0 to 9. Notably, the experimental results demonstrated that this novel framework surpassed the performance of existing techniques in this domain.

Riya Guha et al. [14] introduced an innovative CNN model explicitly designed for identifying offline handwritten Devanagari characters. To arrive at the final architecture, the research involved a comprehensive evaluation of multiple CNN models using publicly accessible datasets of handwritten Devanagari characters and numerals. The resulting architecture is comprised of six distinct layers, and the outcomes of the simulations demonstrated superior performance in character recognition.

Mayur Bhargab Bora et al. [15] implemented a novel technique for HCR, leveraging the fusion of CNN and Error-Correcting Output Codes (ECOC) classifiers. The innovative system employed CNN for the vital task of feature extraction, while ECOC was responsible for character recognition. To identify the most compelling feature extractor, the researchers explored three well-established CNN architectures: LeNet, AlexNet, and ZfNet. The comprehensive experimentation revealed that the ECOC classifiers outperformed the traditional CNN softmax classifier on the basis of accuracy. Among the CNN architectures investigated it was determined that AlexNet was the most appropriate choice for integration with ECOC, enhancing the ability of HCR.

Zhiyuan Li et al. [16] introduced a novel matching network inspired by the way humans learn to write Chinese characters, which establishes connections between template characters and handwritten ones. This novel technique entails substituting the parameters in the softmax regression layer with features derived from template character images. They rigorously assessed the performance of this matching network on the CASIA-HWDB dataset and found that it demonstrated comparable results to existing CNN-based classifiers.

Jyoti Pareek et al. [17] developed an offline Handwriting Character Recognition (HCR) system for the Gujarati language that leveraged artificial intelligence. The proposed approach incorporated a diverse dataset comprised of images contributed by 250 individuals spanning various age groups and professions. The paper detailed the implementation of a supervised classifier approach employing CNN and Multilayer Perceptrons (MLP) for the purpose of recognizing handwritten Gujarati characters. The simulation results revealed an impressive success rate, with CNN achieving a recognition accuracy of 97.21%, significantly outperforming

MLP, which attained a success rate of 64.48%. Shobha Rani et al. [18] introduced an innovative method for identifying handwritten Kannada characters by leveraging transfer learning principles from a pre-existing Devanagari handwritten recognition system. This knowledge transfer process was executed through the adaptation of a DL network architecture, specifically the VGG19 NET. By harnessing a substantial volume of training data, this method effectively addressed the challenging task of recognizing a vast array of 188-character classes. It demonstrated a commendable accuracy rate of approximately 73.51% during validation.

Hossam Magdy Balaha et al. [19] introduced a robust system designed for Arabic AHCR. This system offers the flexibility of choosing between two distinct Convolutional Neural Network (CNN) architectures and operates on a substantial and intricate dataset tailored explicitly for the AHCR task. It also incorporated regularization techniques to explore the impact of the reduction of model complexity and help mitigate overfitting issues. The results obtained indicate that the application of data augmentation techniques played a pivotal role in enhancing testing accuracy while simultaneously reducing the risk of overfitting. Notably, the first CNN architecture surpassed the others in terms of accuracy.

Yu Weng and Chunlei Xia [20] developed an image-processing module tailored to mobile devices, taking into account the distinctive features of Convolutional Neural Networks (CNNs). Initially, a dataset comprising Shui characters was curated, and the CNN model's proficiency in character classification was demonstrated, highlighting its suitability for mobile device deployment. The proposed approach, employing a CNN, has been rigorously validated through comparative analysis against established optical character recognition methods.

A novel framework for a deep CNN with strong recognition performance and the ability to learn deep features for visualization was proposed by Pavlo Melnyk et al. [21]. The bottleneck layers in the suggested model are a unique feature that allows us to maintain its expressiveness while lowering the quantity of multiply-accumulate operations and the necessary storage. According to the simulation results, the suggested strategy is 0.02% more accurate. The proposed model performs well in terms of recognition.

Ferdin Joe John Joseph [22] proposed a local feature-based method employing supervised learning methods to improve the precision of HCR for the Thai alphabet using unsupervised learning for individual characters as a class. Support Vector Machines (SVM) are used in the categorization process. The accuracy would be the proportion of correctly classifying objects in each category. The highest achieved accuracy in the results stands at 74.32%, and it was obtained using 144-bit shape features combined with the uniform pattern Local Binary Pattern (LBP) for feature extraction.

A Bangla HCR system utilizing the MobileNet V1 architecture was proposed by Tapotosh Ghosh et al. [23]. MobileNet has a compact design. It uses depth-wise separable convolution blocks rather than typical convolutional blocks to extract features from input images of 224x224x3. It recognized 231 classes with a 96.46% accuracy rate, 171 compound character classes with a 96.17% accuracy rate, 50 basic character classes with a 98.37% accuracy rate, and 10 numeral character classes with a 99.56% accuracy rate.

An effective KNN-supervised machine learning algorithm was presented by Rohan Sethi et al. [24] for the categorization of handwritten digits. The classifier receives the labelled train data to train its model. A KNN-supervised machine learning technique was employed for classification. The calculation of the supervisory signal that best matches the input vector was done using the Euclidean distance formula. The suggested approach produced outstanding outcomes.

Wenfei Liu et al. [25] provided an overview of the handwriting recognition application and discussed the differences between DL and ML. In this study, the MNIST dataset serves as the sample data, and it elucidates the core principles behind four prominent methods for recognizing handwritten digits. By conducting various simulations and adjustments, the study obtains recognition rates for ANN, SVM, BNN, and CNN. A comparative analysis of the experimental findings reveals that the CNN algorithm has the most significant positive impact on the recognition rate.

3. Materials and Methods

The broad availability of large amounts of annotated data and subsequent architectural advancements have been the primary factors behind the success of DL models. In this paper, a systematic investigation is conducted to enhance the recognition of handwritten text in scanned offline document images. The study introduces a modified architecture, blending CNN with RNN, with a primary focus on optimizing the training process.

Special attention is given to domain-specific data transformations and distortions to capture crucial invariances. The main goal of this study is to establish a framework for handwritten character recognition by integrating a Bi-GRU-based RNN with CNN. Leveraging a large dataset enhances the accuracy of HCR. Feature sequences and recurrent layers are harnessed for the recognition process. The detailed block diagram of the suggested HCR model is illustrated in Figure 4.

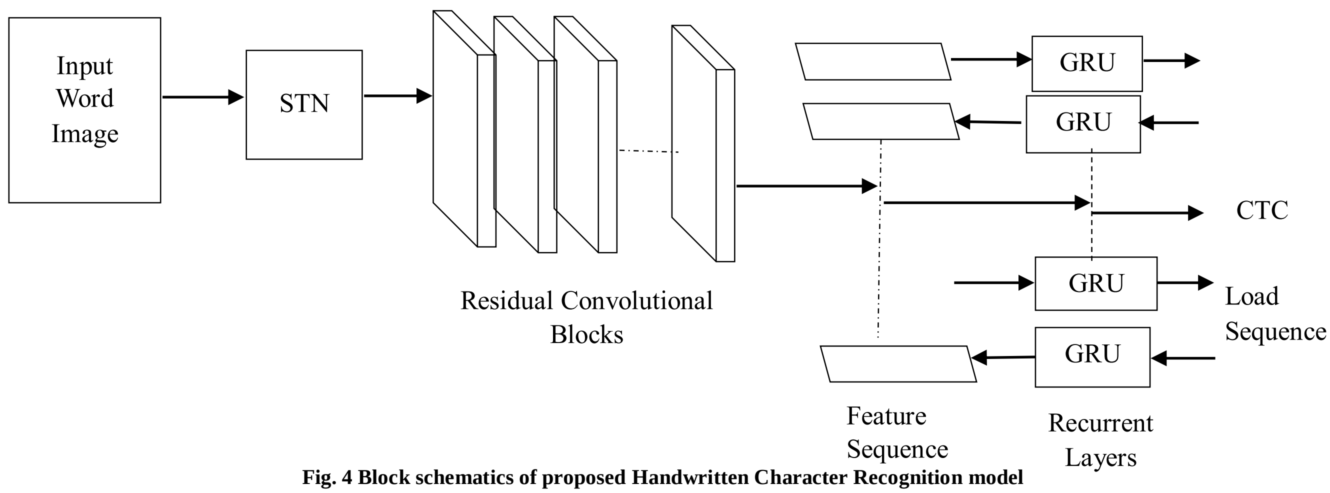


Fig. 4 Block schematics of proposed Handwritten Character Recognition model

Word recognition entails the task of converting handwritten text from an image into computer-readable text. In this study, a hybrid CNN-RNN framework was utilized. This architecture commences with a linear layer for scribing the labels. Subsequently, it incorporates a Spatial Transformer Network (STN), a sequence of residual convolutional blocks, a stacked GRU (Bidirectional GRU), and another linear layer. The STN network functions as a layer that may be trained from beginning to end. Its primary function is to use geometric transformations on the input to fix handwriting deformities brought on by a variety of hand movements. It can be applied to fix a number of geometric transformations, including affine and thin-plate splines.

Here, a series of feature maps are learned using convolutional layers. The stacked GRUs are then given the feature maps as input. Recurrent layers receive a sequence of feature vectors as input from the final convolutional layer that produces a feature map. This method involves converting the predictions made by the recurrent layers into the most likely sequence for the incoming data.

The SoftMax activation creates a probability distribution for class labels at each time step in this process, and the final classification layer uses this probability distribution during testing. The labels with the highest likelihood are put together to make the final sequence, and any extraneous or empty labels are then removed.

3.1. Dataset

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 shown in Figure 5.

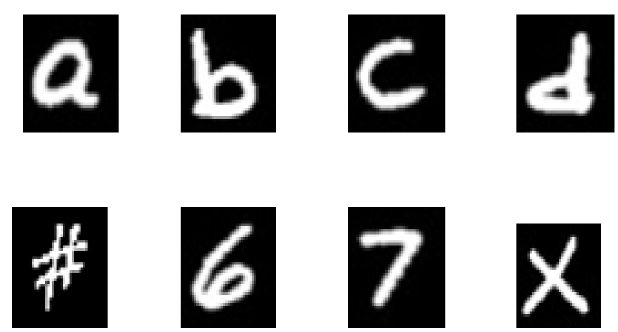


Fig. 5 Sample image from dataset

3.2. Spatial Transformer Network (STN)

For purposes including image modification, alignment, and spatial manipulation, neural networks sometimes include a layer called a spatial transformer layer. It is beneficial for tasks like handwritten character identification since it can direct the network’s attention to crucial areas of the input image and increase accuracy. The basic architecture of STN is illustrated in Figure 6.

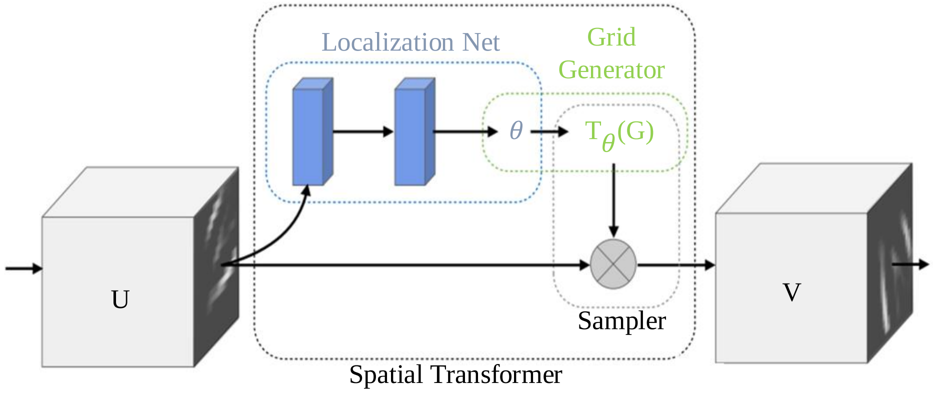


Fig. 6 Spatial Transform Network

The "localization network" is the first component of the spatial transformer layer. This network receives an image as input and outputs a collection of parameters that specify the image's transformation. The values for translation, rotation, scaling, and shearing commonly make up these parameters. Convolutional Neural Networks (CNN) or fully connected networks can be used as the localization network.

The localization network's prediction of the parameters is then utilized to construct a grid. This grid shows the transformation that should be applied to the output image in relation to the input image. It outlines how each output pixel should be sampled from the input image while accounting for the specified transformation. The grid created in the previous phase is used as a sampling grid to gather data from the input image.

According to the grid, each pixel in the output image represents a specific location in the input image. The grid-specified sampling sites are used to build the output image by taking values from the input image. Interpolation methods like bilinear interpolation are used to acquire values at non-integer places. This guarantees a smooth transition that does not introduce artefacts. The altered output image is produced using the sampled values. The image has been modified and is now prepared for additional neural network processing, such as classification for HCR.

3.3. Convolutional Neural Network (CNN)

CNN is well-known for its ability to analyze visual imagery because it is a feed-forward, deep ANN that operates in a similar way to how people perceive it. CNNs use a variety of MLP techniques, which reduces the need for input data preprocessing. Normal neural networks and CNNs function in a reasonably similar manner. Every neuron takes some inputs, does a dot product, and then may choose to follow it with non-linearity. The loss function of CNNs is also present in the Fully Connected layer.

In contrast to the typical matrix multiplication found in neural networks, each unit within a CNN layer functions as a 2-D filter that convolves with the input of that particular layer. This represents a notable departure from other neural network architectures. A typical CNN consists of millions of neurons, which are organized in several hierarchical layers, as depicted in Figure 7.

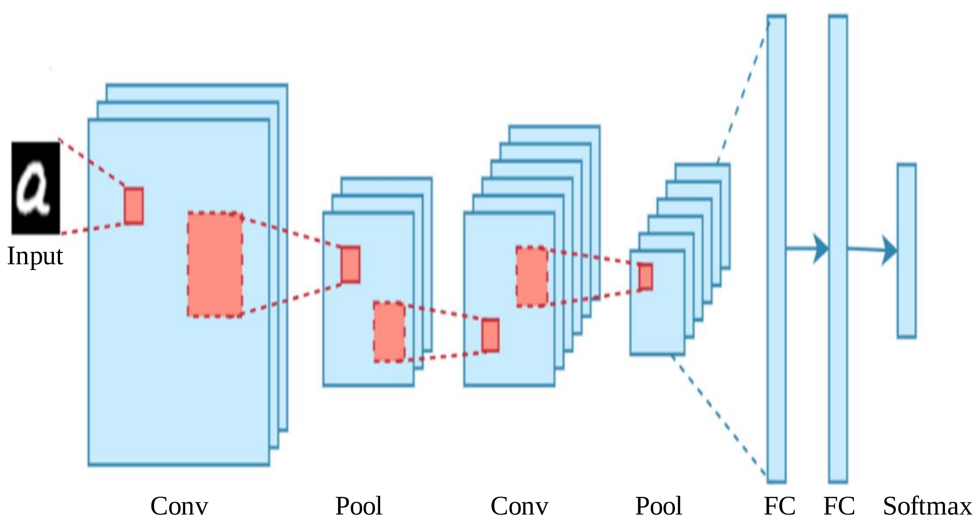


Fig. 7 Basic architecture of CNN

In CNN designs, there are three fundamental layers: convolutional, pooling, and Fully Connected (FC). Sometimes, pooling layers are optional and just partially related to the input images to reduce their size. The fully connected output layer of CNN primarily functions as a classifier. The learnability of the networks is determined by the hidden layers found in the convolutional and FC layers. The number of layers in a CNN determines its depth, and the deeper a layer is, the more information it can extract at a given level. The scaling of the higher-resolution images is facilitated by the neuron connections between the neurons in the hidden layer and those in the next layer. The input layer neurons in CNN are stimulated by visual input. The basic function of the convolution layer is to collect image features, send them into the hidden layers for processing, and retrieve the outcomes through the output layer. Activation functions frequently help in the transfer of essential information for the following layers between hidden levels.

The convolution operation is mathematically expressed, as shown in Equation 1.

$$y(m,n) = \sum_{i=0}^{k-1} \sum_{j=0}^{k-1} x[m+1, n+j].w(i,j) \qquad (1)$$

Where y (m, n) is one data word in the output, x [m, n] is an input data word, and w [m, n] are values of the filters.

3.4. Recurrent Neural Network (RNN)

Recurrent Neural Networks (RNNs) are an extension of feed-forward networks that can handle sequences of varying lengths by utilizing hidden recurrent states connected in a loop. They offer a straightforward approach to processing sequential data in the form of $x_1, x_2, \dots, x_n$. Sequential data often comes with differing lengths or numbers of elements, such as words or values. In an RNN, the output from one state is looped back as input, as illustrated in Figure 8, showing the unfolding of an RNN over three time steps. Each time step has a hidden state. The variables x, y, h are vectors with specific dimensions. In the unfolded portion of Figure 8, h_t signifies the hidden state vector at time-step t. The hidden state acts as memory in the network.

The h_t is determined according to the current input x_t and the previous hidden state h_{t-1} as explained by Equation 2 and 3.

$$h_t = f(x_t, h_{t-1}) \qquad (2)$$

$$h_t = f(Ax_t + Wh_{t-1}) \qquad (3)$$

The output vector at timestep t is calculated only based on hidden state or memory at time-step t. It is obtained as in Equation 4.

$$y_t = SoftMax\,(Bh_t) \qquad (4)$$

3.4.1. Gated Recurrent Unit (GRU)

An essential gating function is offered by Gated Recurrent Units (GRU). It has reset and updated gates, respectively. The reset gate computes a sigmoid function over input from the previous hidden state and the current input. The values of the update gate and reset gate at time step t are calculated as Equations 5 and 6.

$$u_t = \sigma(A^u x_t + W^u h_{t-1}) \qquad (5)$$

$$r_t = \sigma(A^r x_t + W^r h_{t-1}) \qquad (6)$$

How much of the information from the previous phases should be kept depends on the value of the sigmoid operation. The reset value will range from 0 to 1. The reset gate is responsible for clearing the hidden state vector or memory and storing new data if the value is 0. The update gate enables the controlled transfer of hidden state vectors. In GRU, before computing the hidden state h_t, an intermediate hidden state $\widehat{h_t}$ is computed as in Equation 7.

$$\widehat{h_t} = \tanh(Ax_t + r_t \circ Wh_{t-1}) \qquad (7)$$

A zero value in r_t forces to ignore the previous hidden state information or memory. The hidden state (h_t) at time-step t is computed as in Equation 8.

$$h_t = u_t \circ h_{t-1} + (1-u_t)\widehat{h_t} \qquad (8)$$

The illustration of GRU at hidden state computation at h_t is shown in Figure 9. The value in the update gate u_t determines how much of the previous hidden state information needs to be carried to compute the current hidden state h_t.

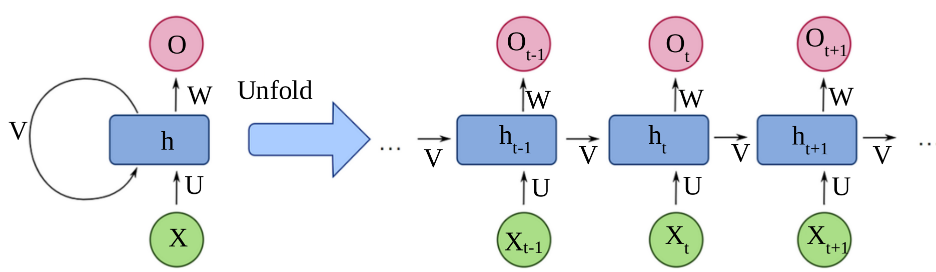


Fig. 8 Unfolding of an RNN with three time-steps

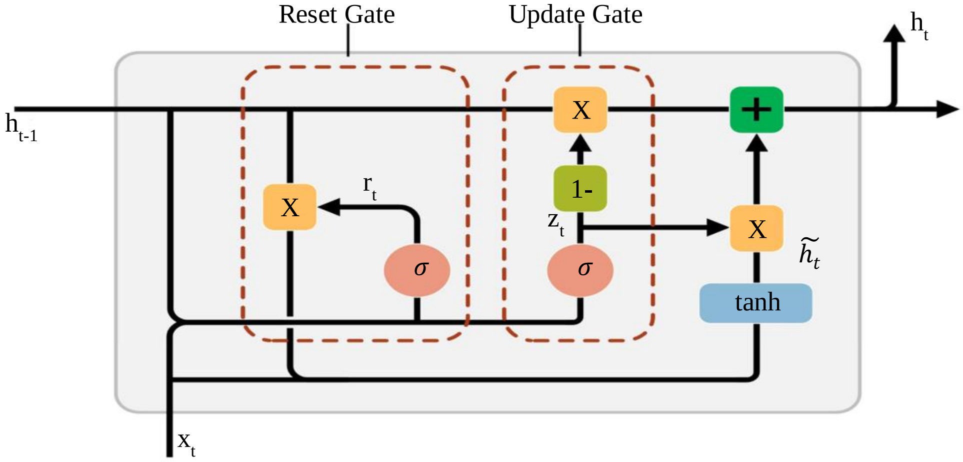


Fig. 9 Visualization of Gated Recurrent Unit

3.4.2. Bidirectional Gated Recurrent Unit

A sequence processing model called the Bidirectional Gated Recurrent Unit (Bi-GRU) combines the most significant features of the BI-RNN and GRU. The forward and backward hidden layers of BI-RNN can use both past and future data at the same time. As illustrated in Figure 10, which depicts the architecture of the BI-GRUs network, the forward and backwards hidden layer nodes of the RNN are swapped out for GRU cells to achieve the bidirectionality of the RNN.

The sequential data is fed into both the forward and the backward GRU layers simultaneously in this model, which uses two hidden layers of GRU to process information simultaneously in both directions. This enables the model to work with the present data while learning from both earlier and later data. Thus, the current data reflects the combined influence of both past and future information. Specifically, x_t serves as the input to the GRU while h_f represents the output of the forward GRU layer, and h_b signifies the output of the reverse GRU layer. The resulting outputs, denoted as $[h_1,\ h_2,\ h_3,\ h_4]$, are produced through the collaborative operation of these two reverse GRU layers.

The CTC loss seeks to calculate the likelihood of a target character sequence given the input image. It accounts for all potential alignments, including insertions and repetitions, between the input image and the target sequence. The forward-backwards algorithm is a dynamic programming technique used by the CTC loss to calculate the probability of the desired sequence. The forward pass and the backward pass are two passes that are involved. The forward pass calculates the likelihood of producing the target sequence given the input. When given the target sequence, backward pass calculates the conditional probability of producing the input. The probabilities acquired in each pass are then added to determine the CTC loss. In order to optimize the probability of the target sequence, the input and target sequences should be aligned in the model to minimize this loss.

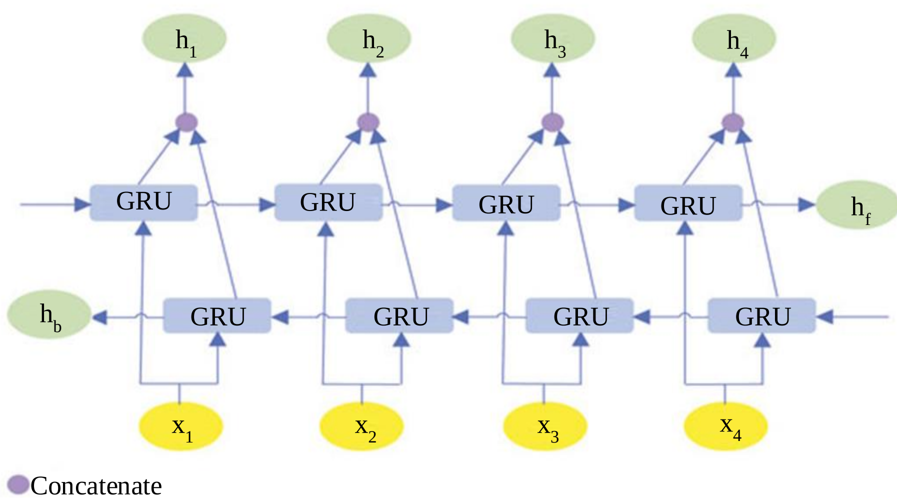


Fig. 10 Structure of Bidirectional Gated Recurrent Unit

The model is initiated by employing a 2D convolutional layer, followed by batch normalization, activation using ReLU, and max-pooling layers to retrieve relevant features from the input images. Dropout layers follow these convolutional layers to reduce overfitting. The output from the CNN part is then reshaped and passed through two bidirectional Gated Recurrent Unit (GRU) layers to capture sequential dependencies in the data.

Finally, the model's output is obtained using a softmax activation layer, with a total of 35 classes, making it suitable for a classification task. This design integrates the feature extraction abilities of CNNs with the sequential modelling abilities of RNNs, making it a suitable choice for tasks that demand a comprehensive grasp of both spatial and temporal aspects of the data.

4. Results and Discussion

4.1. Hardware and Software Setup

The implementation faced several challenges that necessitated changes to the development environment to meet the project's requirements. Experiments and simulations were conducted using Python, as the image data was provided in the form of .jpg files. To streamline the process, Python scripts were used to facilitate image downloads, and all images were relocated to Google Drive for easy access during the model-building phase. Additionally, image processing and exploratory work were carried out on a high-end workstation equipped with 32GB of RAM, an 8GB Nvidia Graphics Processing Unit, and an Intel i7 Processor.

The model, a combination of CNN and GRU, contains a total of 1,868,864 trainable parameters, which are the learnable weights and biases that the model adjusts during training to make accurate predictions. Adam, a well-liked optimization technique, was selected as the optimizer for this model. Adam adjusts these parameters based on the gradients of the loss function. The step size at which the parameters of the model are changed during training is determined by the learning rate for the optimizer, which is set at 0.001.

Categorical cross entropy was selected as the loss function because it measures the difference between the class probabilities that are predicted and those that are actually observed, directing the model to minimize this difference. In order to balance computing efficiency and learning stability, data processing is done in 64-sample batches during training. Each epoch throughout the 50-epoch training procedure represents a complete iteration through the whole training dataset, enabling the model to enhance performance progressively. These parameters collectively shape the model's learning process, determining how it adapts and generalizes to the given data, ultimately affecting the quality of its predictions. The model configuration is tabulated in Table 1.

4.2. Performance Evaluation

An accuracy plot is a visual representation used in performance evaluation to assess the correctness of predictions made by a model. It typically shows how the model's accuracy, or the proportion of correctly classified instances, changes over time or with varying parameters. This plot is an essential tool in machine learning and data analysis, enabling practitioners to understand how the model's performance evolves as it is trained or fine-tuned. In a typical accuracy plot, the horizontal axis may represent the number of training epochs, different parameter values, or other relevant factors. At the same time, the y-axis shows the accuracy achieved by the model. A rising trend indicates improving performance, while fluctuations or plateaus may highlight issues or diminishing returns in model training. Accuracy plots help in selecting the best-performing model, monitoring convergence, and making informed decisions about model adjustments. The accuracy plot of the suggested HCR model is visualized in Figure 11.

A loss plot is a crucial component of performance evaluation in machine learning. It provides a visual representation of how well a model is learning from the training data over time. In the plot, a declining loss curve indicates that the model is learning and improving its predictive accuracy. Conversely, if the loss plateaus or increases, it may signify overfitting, underfitting, or other issues. By monitoring the loss plot, practitioners can assess model performance, make necessary adjustments, and determine when the model has reached a satisfactory level of learning, thus aiding in the decision to stop or continue training. The loss plot of the suggested model is visualized in Figure 12.

Table 1. Model configuration

Model Parameters	Proposed Model (CNN-Bi GRU)
Trainable Parameters	1,868,864
Optimizer	Adam
Learning Rate	0.001
Loss Function	Categorical Cross Entropy
Batch Size	64
Number of Epochs	50

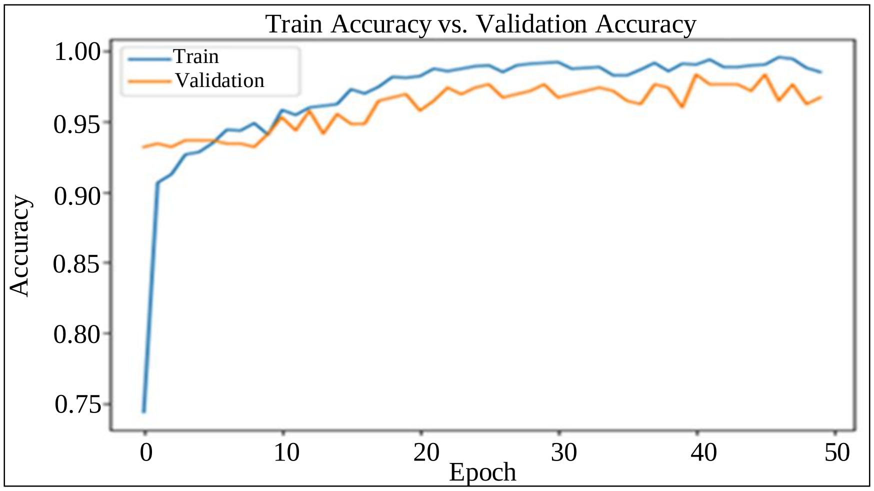


Fig. 11 Accuracy plot of proposed model

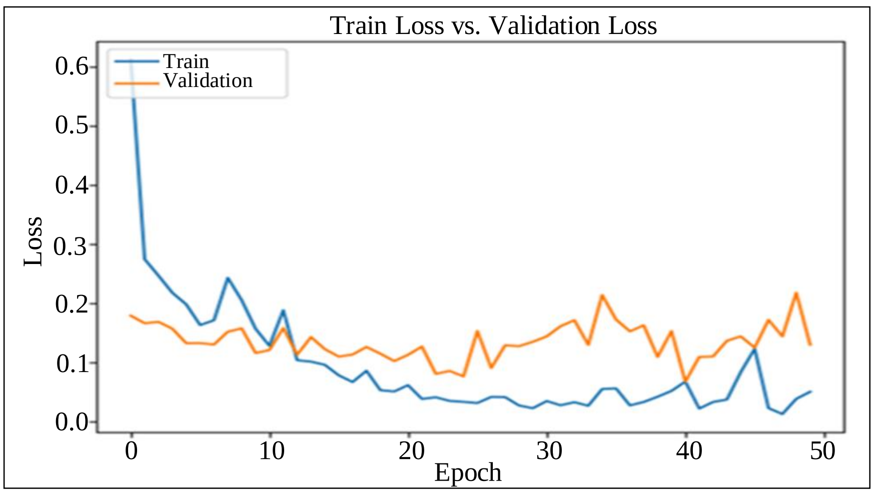


Fig. 12 Loss plot of the proposed model

Accuracy is a fundamental metric used in performance evaluation to assess the effectiveness of a model or system in making correct predictions. In classification tasks, accuracy indicates how often the model's predictions match the actual true values, and it is typically expressed as a percentage. While accuracy provides a straightforward and intuitive measure of overall correctness, it may not always be the most suitable metric, especially when dealing with imbalanced datasets where one class significantly outnumbers the others. Accuracy is mathematically expressed as in Equation 9.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \tag{9}$$

The proposed hybrid CNN-RNN model with a Bi-GRU model provided an average accuracy of 96.72%. The images from the validation folder are used to predict the characters in the handwritten word. The sort contour's function is used to get the correct order of individual characters for correct output extraction. Some of the prediction outputs utilizing the suggested approach are tabulated in Table 2.

Table 2. Prediction of characters

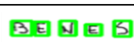
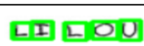
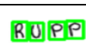
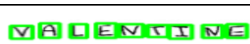
Input Image	Resulting Text
	BENES
	LILOU
	JAFFEUX
	RUPP
	VALENTINE

Table 3. Performance comparison

Reference No.	Author and Year	Title	Methodology	Accuracy
[26]	Anita Pal and Dayashankar Singh (2010)	Handwritten english character network recognition using neural network	Fourier descriptors with backpropagation	94%
[27]	Md. Mahbubar Rahman et al. (2015)	Bangla handwritten character recognition using convolutional neural network	Convolutional Neural Network	85.36%
[28]	Aiquan Yuan et al. (2012)	Offline handwritten english character recognition based on convolutional neural network	Convolutional Neural Network	93.7%
[29]	Ahmed El-Sawy et al. (2017)	Arabic handwritten characters recognition using convolutional neural network	Convolutional Neural Network	94.9%
[30]	N. Shanthi E.K. Duraiswamy (2017)	A novel SVM-based handwritten Tamil character recognition system	Support Vector Machine	82.04%
Proposed Method			CNN-RNN with Bi-GRU	96.72%

4.3. Performance Comparison

In the realm of handwritten character recognition, various methodologies have been explored for different languages. Table 3 showcases diverse approaches and their corresponding accuracies, emphasizing the effectiveness of the proposed CNN-RNN with the Bi-GRU method for handwritten character recognition.

5. Conclusion

HCR is a critical and multifaceted area of research and application with broad-reaching implications. In a world increasingly reliant on digital information, HCR offers several significant conclusions and insights. HCR finds applications in diverse fields, ranging from digitizing historical documents and cursive handwriting to automatic address reading on postal envelopes, form processing, and aiding individuals with disabilities by converting handwritten text into speech. Recognizing handwritten characters poses substantial challenges due to the immense variability in individual writing styles, letter shapes, and the presence of noise, which can arise from imperfect writing or scanning conditions.

Adapting to this variability is crucial for recognition systems. Advances in DL techniques, especially CNNs and RNNs, have significantly enhanced the accuracy and performance of HCR systems. These models can capture complex patterns and dependencies in handwritten data. This paper proposes an effective HCR model using CNN-RNN with Bi-GRU. A handcrafted, handwritten character dataset has been developed. The suggested model attained an accuracy of 96.72%. The CNN component is crucial for retrieving spatial features and patterns from the input images, effectively capturing information related to the shapes and structures of characters. The RNN with Bi-GRU layers is instrumental in modelling sequential dependencies, which is essential for recognizing the temporal aspects of handwriting, especially in situations where characters are connected or vary in writing styles. Future HCR models should be considerably more sophisticated and potent as data becomes more readily available and technology develops. This hybrid approach has paved the way for automation, enhanced data processing, and improved user experiences in numerous industries.

Acknowledgements

I would like to express my sincere gratitude to all those who contributed to the completion of this research paper. I extend my heartfelt thanks to my supervisor, P. Ranjit Jeba Thangaiah, for their invaluable guidance and unwavering support throughout the research process. I extend my thanks to my family, my colleagues and fellow researchers for their encouragement and understanding during the demanding phases of this work.

References

[1] Charles A. Perfetti, and Susan Dunlap, *Learning to Read: General Principles and Writing System Variations*, Learning to Read across Languages, Routledge, pp. 25-50, 2008. [Google Scholar] [Publisher Link]

[2] Umapada Pal, Ramachandran Jayadevan, and Nabin Sharma, "Handwriting Recognition in Indian Regional Scripts: A Survey of Offline Techniques," *ACM Transactions on Asian Language Information Processing*, vol. 11, no. 1, pp. 1-35, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[3] Paul Westwood, and Gary Crew, "The Future of Handwriting," *Australian Association of Writing Programs*, vol. 22, no. 52, pp. 1-10, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[4] M. Swamy Das, D. Sandhya Rani, and C.R.K. Reddy, "Heuristic Based Script Identification from Multilingual Text Documents," *2012 1st International Conference on Recent Advances in Information Technology (RAIT)*, Dhanbad, India, pp. 487-492, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[5] J. Mantas, "An Overview of Character Recognition Methodologies," *Pattern Recognition*, vol. 19, no. 6, pp. 425-430, 1986. [CrossRef] [Google Scholar] [Publisher Link]

[6] Jamshed Memon et al., "Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)," *IEEE Access*, vol. 8, pp. 142642-142668, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[7] K. Yoshida, and H. Sakoe, "Online Handwritten Character Recognition for a Personal Computer System," *IEEE Transactions on Consumer Electronics*, vol. CE-28, no. 3, pp. 202-209, 1982. [CrossRef] [Google Scholar] [Publisher Link]

[8] N. Arica, and F.T. Yarman-Vural, "An Overview of Character Recognition Focused on Off-Line Handwriting," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 31, no. 2, pp. 216-233, 2001. [CrossRef] [Google Scholar] [Publisher Link]

[9] Amitha Mathew, P. Amudha, and S. Sivakumari, "Deep Learning Techniques: An Overview," *Advanced Machine Learning Technologies and Applications*, pp. 599-608, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[10] K.R. Chowdhary, "Natural Language Processing," *Fundamentals of Artificial Intelligence*, pp. 603-649, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[11] Yasir Babiker Hamdan, and A. Sathesh, "Construction of Statistical SVM Based Recognition Model for Handwritten Character Recognition," *Journal of Information Technology and Digital World*, vol. 3, no. 2, pp. 92-107, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[12] I. Khandokar et al., "Handwritten Character Recognition Using Convolutional Neural Network," *Journal of Physics: Conference Series*, vol. 1918, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[13] Saleh Albahli et al., "An Improved Faster-RCNN Model for Handwritten Character Recognition," *Arabian Journal for Science and Engineering*, vol. 46, no. 9, pp. 8509-8523, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[14] Riya Guha et al., "Devnet: An Efficient CNN Architecture for Handwritten Devanagari Character Recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 34, no. 12, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[15] Mayur Bhargab Bora et al., "Handwritten Character Recognition from Images Using CNN-ECOC," *Procedia Computer Science*, vol. 167, pp. 2403-2409, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[16] Zhiyuan Li et al., "Deep Matching Network for Handwritten Chinese Character Recognition," *Pattern Recognition*, vol. 107, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[17] Jyoti Pareek et al., "Gujarati Handwritten Character Recognition from Text Images," *Procedia Computer Science*, vol. 171, pp. 514-523, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[18] N. Shobha Rani et al., "Deep Learning Network Architecture-Based Kannada Handwritten Character Recognition," *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, pp. 213-220, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[19] Hossam Magdy Balaha et al., "A New Arabic Handwritten Character Recognition Deep Learning System (AHCR-DLS)," *Neural Computing and Applications*, vol. 33, pp. 6325-6367, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[20] Yu Weng, and Chunlei Xia, "A New Deep Learning-Based Handwritten Character Recognition System on Mobile Computing Devices," *Mobile Networks and Applications*, vol. 25, pp. 402-411, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[21] Pavlo Melnyk, Zhiqiang You, and Keqin Li, "A High-Performance CNN Method for Offline Handwritten Chinese Character Recognition and Visualization," *Soft Computing*, vol. 24, no. 11, pp. 7977-7987, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[22] Ferdin Joe John Joseph, "Effect of Supervised Learning Methodologies in Offline Handwritten Thai Character Recognition," *International Journal of Information Technology*, vol. 12, no. 1, pp. 57-64, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[23] Tapotosh Ghosh et al., "Bangla Handwritten Character Recognition Using MobileNet V1 Architecture," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 6, pp. 2547-2554, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[24] Rohan Sethi, and Ila Kaushik, "Hand Written Digit Recognition Using Machine Learning," *2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT)*, Gwalior, India, pp. 49-54, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[25] Wenfei Liu, Jingcheng Wei, and Qingmin Meng, "Comparisions on KNN, SVM, BP and the CNN for Handwritten Digit Recognition," *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*, Dalian, China, pp. 587-590, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[26] Anita Pal, and Dayashankar Singh, "Handwritten English Character Recognition Using Neural Network," *International Journal of Computer Science & Communication*, vol. 1, no. 2, pp. 141-144, 2010. [Google Scholar] [Publisher Link]

[27] Md. Mahbubar Rahman et al., "Bangla Handwritten Character Recognition Using Convolutional Neural Network," *International Journal of Image, Graphics and Signal Processing*, vol. 7, no. 8, pp. 42-49, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[28] Aiquan Yuan et al., "Offline Handwritten English Character Recognition Based on Convolutional Neural Network," *2012 10th IAPR International Workshop on Document Analysis Systems*, Gold Coast, Australia, pp. 125-129, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[29] Ahmed El-Sawy, Mohamed Loey, and Hazem EL-Bakry, "Arabic Handwritten Characters Recognition Using Convolutional Neural Network," *WSEAS Transactions on Computer Research*, vol. 5, no. 1, pp. 11-19, 2017. [Google Scholar] [Publisher Link]

[30] N. Shanthi, and K. Duraiswamy, "A Novel SVM-Based Handwritten Tamil Character Recognition System," *Pattern Analysis and Applications*, vol. 13, pp. 173-180, 2010. [CrossRef] [Google Scholar] [Publisher Link]