Journal of Computer Science

Original Research Paper

Online Handwritten Physics Expression Recognition Using a CRNN-LSTM Approach

[1]Ujwala Pritam Kolte and [2]Sachin Arun Naik

[1]*Faculty of Computer Studies, Symbiosis International (Deemed University), Pune, India*
[2]*Department of Computer Science and Applications, Dr. Vishwanath Karad MIT World Peace University, Kothrud, Pune, India*

Article history
Received: 26-07-2024
Revised: 07-10-2024
Accepted: 25-11-2024

Corresponding Author:
Sachin Arun Naik
Department of Computer Science and Applications, Dr. Vishwanath Karad MIT World Peace University, Kothrud, Pune, India
Email: sachinnaik13@gmail.com

Abstract: Optical Character Recognition (OCR) is crucial for identifying and extracting information from scientific documents. However, recognizing online handwritten physics expressions remains a relatively unexplored area of research. In the existing literature, most of the researchers worked on CNN classification to identify mathematical expressions. The traditional Convolution Neural Network (CNN) has limitations for extracting sequential data. This study proposes a CRNN (Convolution Recurrent Neural Network)-LSTM (Long Short-Term Memory) model that uses sequence a to-sequence approach for recognition of online handwritten physics expressions to enhance the feature extraction process for the sequential data which overcomes the limitations of CNN. A dataset was constructed from 100 users using a Java-based interface, covering five commonly used types of physics expressions: (1) Electric flux; (2) Maxwell's equations; (3) Inductance; (4) Pointing vector, and (5) Moment of inertia. This dataset identifies 25 unique Physics symbols. The Online Handwritten Physics Expression (OHPE) dataset comprises a total of 500 physics expressions, including operators, special symbols, alphabets, and numbers. The proposed architecture of CRNN-LSTM architecture consists of convolution layers followed by LSTM layers for sequence processing. CTC (Connectionist Temporal Classification) loss function is used to train the model, which is particularly used to predict the sequence of symbols within the physics expressions. The performance evaluation has been carried out and the accuracy of symbol level and expression level prediction is reported. In the final stage the trained model is evaluated to make the predictions on the unseen test set images with the recognition accuracy of 96.36% at the expression level and 98.10% at the symbol level.

Keywords: Physics Expression Recognition, Segmentation, Convolutional Neural Network, Recurrent Neural Network, Long Short-Term Memory Networks

Introduction

Several handwriting tools are available nowadays, but still, end-to-end recognition of physics or mathematical expressions is an area that needs exploration. Smartphones and tablets are now used on a day-to-day basis, especially in the teaching-learning environment, in which handwritten notes especially in context with mathematical symbols challenge many issues to the users. Students or users expect the hand-written tool to write and store the physics or mathematical expressions into their devices as handwritten notes which makes them convenient instead of using 'insert-expression' operations given in many document editing software. Specially handwritten mathematical expression recognition is difficult due to variations in human handwriting, ambiguous set of symbols exists within the dataset, which leads to less accuracy of prediction of correct symbols.

However, there are many challenges in handwriting which include skewness of data, connected symbols, operators and alphabets, and distorted symbols. Segmentation of connected symbols and characters is a challenging task due to connected, overlapped symbols or operators. This study proposes a CRNN-LSTM approach for the recognition of Online Handwritten Physics Expressions (OHPE) at the expression level and symbol level. In this study, a dataset was collected from 100 different users. There are 5 different types of most commonly used physics expressions namely:

1) Electric flux
2) Maxwell’s equations
3) Inductance
4) Pointing vector
5) Moment of inertia

The dataset consists of 500 unconstrained online handwritten physics expressions which are used for training validation and testing purposes to achieve better accuracy. This study uses the CRNN-LSTM classification technique to recognize the OHPEs. Only using CNN has the limitations of requiring more extensive data, whereas RNN can handle more extensive input. This study proposes an adaptive approach by integrating CNN and RNN, in which the dataset is sequentially trained with CNN and RNN, and further enhanced features are given as input to LSTM to obtain better accuracy for the sequential input. In this study, the CTC (Connectionist Temporal Classification) network is used along with RNN to predict the probability. The proposed model consists of various steps which include data loading, pre-processing, labeling, CRNN layers, Bi-LSTM layer sequence modeling, model training, model validation, model testing, and inference and recognition. The segmentation process involves two steps, expression segmentation and text recognition. The overall model consists of CNN, RNN and CTC layers, which automatically extract features at the expression level. The CNN feature sequence consists of 256 features, further RNN extracts the relevant information feature sequence. RNN output has been provided to the CTC layer to decode the output text. This study introduces innovative methods in two significant areas: (1) The integration of a CRNN-LSTM classification approach and (2) A segmentation-free method for recognizing online handwritten physics expressions.

Literature Review

Lecun *et al*. (2015; 1998) in their research stated that many classifiers like SVM, K-NN, CNN, Random Forest, HMM, and Naïve Bayes are used by researchers in the field of mathematical symbol recognitions. CNN shows promising results due to its ability to generate automated hierarchical features directly from images. SVMs have been widely used for handwritten symbol recognition tasks, particularly when combined with suitable feature representations such as HOG or Scale-Invariant Feature Transform (SIFT) descriptors shown in Cortes and Vapnik, (1995); Karatzas *et al*. (2013); Rabiner (1989); Bunke *et al*. (2004) used HMMs for recognizing temporal sequences of handwritten symbols, especially for cursive handwriting recognition where the order of strokes may convey important information. Graves *et al*. (2009); Hochreiter and Schmidhuber, (1997) demonstrated RNNs, especially variants like LSTM networks, have been utilized for recognizing sequential patterns in handwritten mathematical expressions. Dietterich (2000); and Diem *et al*. (2013) use ensemble methods such as random forests or AdaBoost can be effective in combining multiple weak classifiers to improve overall recognition performance. Geetha *et al*. (2021) proposed recognition of words and characters of different lengths to create sentences. The training utilizes the IAM and RIMES datasets, with CNN employed for feature extraction, encompassing visual attributes such as horizontal gradients and orientation features. The model is structured with a CNN phase followed by an RNN phase, where extracted features are input to an LSTM-RNN sequential learner to generate the final digital output. The paper compares the word accuracy and letter accuracy of CNN, RNN, and the proposed H2TR (CNN-RNN) model, achieving respective accuracies of 96.31; 87.10, and 98.14% for words, and 98.31; 95.70 and 99.35% for letters. Safarzadeh and Jafarzadeh (2020) proposed a Persian handwritten word recognizer is presented, incorporating CNN and RNN based on a sequence labeling method. To eliminate the segmentation step, a CTC loss function is utilized. The proposed method is tested on the IFN/ENIT datasets, which include Persian and Arabic samples. This model evaluates the recognition of words and numbers in Persian and Arabic languages, achieving a 99.43% recognition rate for digits and 91.15% for words. Bastas *et al*. (2020) proposed the air writing recognizer to recognize digits from 0-9 using LSTM methods. Three architectures CNN, LSTM, and TCN used to compute the performance of the model. Jain *et al*. (2021) used the CNN-LSTM approach integrated with CTC to recognize medical prescriptions, in this the CTC converts a sequence of characters using the LSTM layer to characters including the use of the Bi-LSTM layer to compute probability. This approach uses a manually created dataset. Sanap *et al*. (2023) utilized deep neural networks and recurrent neural networks to efficiently manage the complexities of handwritten text. After the application of preprocessing techniques like noise removal, normalization, and resizing, relevant features are extracted using CNN. Extracted features are then passed to a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) units to capture temporal dependencies and context. The proposed model is trained using the IAM dataset which contains handwritten English sentences. Ma (2023) presented a solution for handwritten text recognition on the IAM dataset using a hybrid CNN-RNN model. To evaluate the impact of different models, a single-method CNN model is also employed for comparison. Visualization techniques are utilized to illustrate the processing of the input image at each stage, enhancing the understanding of the model's structure and component operations. The results show a significant improvement with the CNN-RNN fusion model, reducing the Word Error Rate (WER) to -12.04% and the Character Error Rate (CER) to -5.13%, compared to the CNN model.

Nurseitov *et al*. (2020) investigated two handwriting recognition models for Kazakh and Russian. The first uses CNNs for feature extraction and an MLP for word classification, while the second, SimpleHTR, combines

CNN and RNN layers. They also introduce a new model, the Bluechet and Puchserver model, for comparison. Their dataset includes handwritten names of countries and cities from 42 Cyrillic words, written over 500 times in various styles, and the Handwritten Kazakh and Russian (HKR) database. The recognition accuracy is 55.3% for the CNN model and 57.1% for SimpleHTR. Albahli *et al.* (2021) use a Faster-RCNN for Handwritten Digit Recognition (HDR), comprising three steps. First, regions of interest are obtained by developing annotations. Next, deep features are computed using DenseNet-41. Finally, the digits are classified into ten classes using a regression and classification layer. The performance of the proposed method is evaluated on the standard MNIST database. The authors compare their method with recent HDR methods, achieving a 99.78% accuracy, the highest among all compared methods. Saini *et al*. (2023) proposed a CRNN and LSTM approach to recognize Gurmukhi characters and performance evaluation is carried out at word level. The dataset includes 175 writers with Dell Latitude XT-3 tablet PC, with a stroke accuracy of 98.67%. Hemanth *et al.* (2021) have proposed the handwritten text recognition model combining CNN, RNN, and CTC layers in which CNN extracts 256 features per time-step, the RNN processes sequence information, and the CTC layer decodes the text. Trained on the IAM word image dataset, the model achieved 98% accuracy on both the IAM database and custom handwritten samples. Zhan *et al.* (2021) introduced an RNN-free architecture for recognizing digit strings, leveraging a CNN and CTC. This architecture includes a feature extractor and a Softmax output layer with a decoder. They evaluated this approach on three established benchmarks for handwritten digit sequence recognition: ORAND-CAR-A, ORAND-CAR-B, and CVL HDS. The authors assert that their model offers advantages in terms of speed and reduces the overall model size. Chammas and Mokbel (2021) proposed an innovative method for fine-tuning HR systems using Temporal Dropout (TD), which randomly drops information at various positions within sequences. The approach is tailored for handling long sequences typically encountered in RNNs and LSTM networks. The effectiveness of this method is evaluated on the Rodrigo and READ16 datasets.

In a summary of the literature, the major research challenges in recognition of online handwritten mathematical or physics expressions are the complexity of expressions, data variation, and size, the segmentation-free handwriting recognition to avoid the low recognition rates due to variability and ambiguity within the mathematical symbols used in the expressions. The traditional models require the segmentation approach to input the mathematical symbols.

Materials and Methods

To conduct this study, dataset preparation is a major challenge. The online handwritten physics expression has been taken as input through a java-based user interface Kolte *et al*. (2024). The most commonly used five categories of expressions are electric flux, Maxwell's equation, inductance, pointing vector, and moment of inertia. The data has been collected from the different 100 writers of age group 16-24 particularly students, where each writer has written 5 categories of Physics expressions, a total of 500 different physics expressions of 5 categories each, have been taken as the input for the training, testing, and validation purpose. The dataset has been prepared using five categories of expression. Table (1) shows five different types of physics expressions used in experimentation.

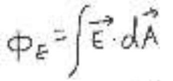
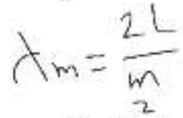
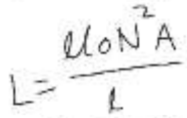
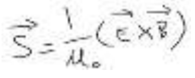
The proposed method consists of several steps given in Fig. (1). The proposed methodology provides the solution to segmentation-free recognition of handwritten physics expressions, operators, and alphabets. The CRNN-LSTM network has the advantage of processing sequential input which is necessary with respect to physics expressions as it has dependency among the sequence of variables, operators, and characters used in physics expressions.

The preprocessing step involves converting an image to grayscale and reshaping the HPE image to 300 widths and 70 heights. The reshaped dimensions are computed using the average size of every sample collected. The image normalization has been carried out to the range 0 to 1. The processed grayscale image is then converted into the binary image, where the contours of the binary image are detected and applied with bounding boxes as shown in Fig. (2).

For the data processing, the following steps are applied:

1. Converting an image into a grayscale image
2. Converting a grayscale image into a binary image using the Otsu method
3. Removing noise from the image by using a median filter
4. Selecting ROI by cropping the image

Table 1: Physics expression types

Expression name	Online handwritten expression
Electric flux	
Maxwell's equations	
Inductance	
The pointing vector	
Moment of inertia	

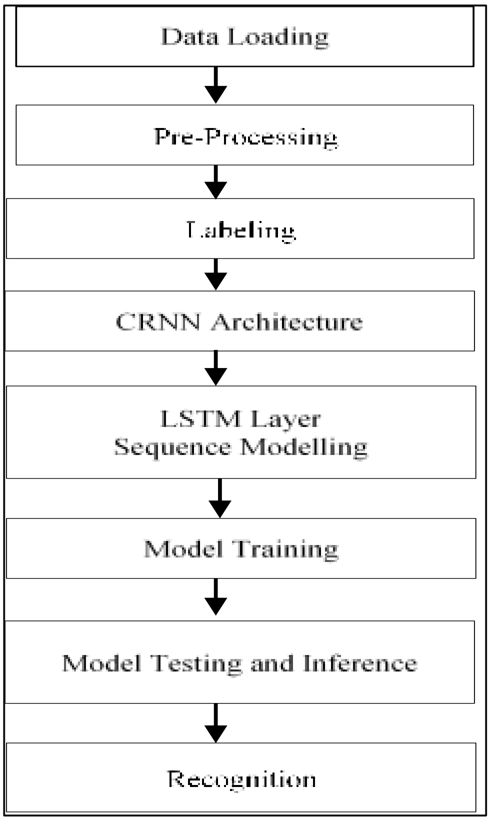


Fig. 1: Overall CRNN-LSTM recognition model architecture

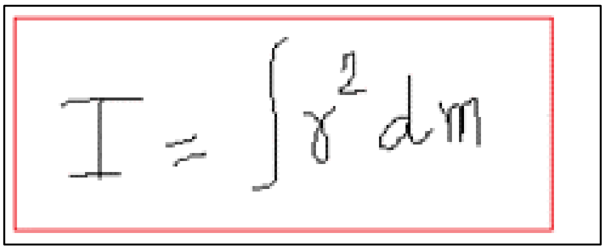


Fig. 2: Bounding box over physics expression

Table 2: OHPE labels

Expression category	Class label	No. of OHPE expressions
Electric flux	EF	100
Maxwell's equations	ME	100
Inductance	IDC	100
The pointing vector	PV	100
Moment of inertia	MI	100
Total		500

The bounding box is applied to the expression level. The subsequent step involves labeling the physics expressions according to the various categories outlined in Table (2).

The limitation of convolution neural network works at the character or symbol level as to one recognition system. In physics expression symbols are varied based on the category of expressions. The correct segmentation of the symbols and operators used in the physics expressions is challenging due to connected symbols, skewness in the expressions, and also due to overall structural representation. There is a need to concentrate on the recognition of complete expression which consists of a group of characters. To address the challenge of one input as a complete expression and many outputs in terms of a varied number of symbols within the physics expressions. So, the classical CNN is not fit for the recognition of handwritten text. In this scenario, Recurrent Neural Networks (RNNs) are the most appropriate choice because they possess a single input and generate multiple outputs (Graves, 2012b). RNNs handle sequences of vectors by iteratively applying a recurrent formula at each step. In Fig. (3), *p* is the input sequence and *q* is the output sequence.

The overall CRNN-LSTM model is built as per Fig. (4). The proposed architecture for the recognition of OHPEs consists of CNN, features which are reshaped to 32*8, i.e. 256 features that are further processed by the RNN+LSTM classification method. CNN networks are used for handwritten image classification. The input expression image is resized to a height of 80 pixels and a height of 256 pixels. The image array is provided to the CNN classification which produces 256 features. CNN is used to extract more hidden patterns by using convolution and pooling layers and produces more useful features by using down down-sampling method. The input handwritten physics expression image is provided as an input to CNN.

Equation (2) shows the convolution operation for the input OHPE image.

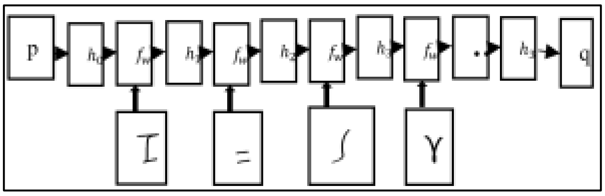


Fig. 3: Inner process in RNN for processing OHPE symbols in sequence

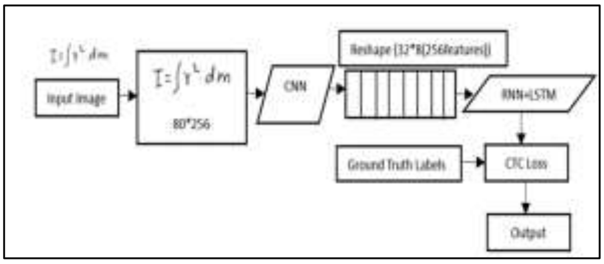


Fig. 4: Proposed CRNN-LSTM model

$$h_{i,j} = \sigma\left(\sum_{m=0}^{k-1} w_m x_{i,j+m} + b\right) \quad (2)$$

In Eq. (2) $h_{i,j}$ is the output of the convolution operation, Wm is the weights, $x_{i,j+m}$ is the input, b is the bias and σ is the activation function. To process the handwritten OHPE image, CNN with three convolution layers is used which is followed by a max-pooling layer for downs-sampling.

The output is flattened and provided to a fully connected layer with 256 neurons which in turn produces 256 features. The first convolution layer uses 32 filters with kernel size 3×3 and the Rectified Linear Unit (ReLU) activation function operating on grayscale images of size 80×256. Subsequently, spatial dimensions are halved by max-pooling layers, this process is repeated by applying 64 and 128 filters, which enhance the feature extraction process to retrieve intrinsic patterns. The ReLU activation function condenses the extracted features into a compact representation. The CRNN-LSTM (Convolution Recurrent Neural Network with Long Sort long-term memory) is an influential architecture to recognize the handwritten text in which the CNN extracts the special features of the symbols used in OHPE whereas RNN spatially LSTM preserves the temporal dependencies in the sequence. Each of the LSTM cells consists of a cell state Ct, which serves as the memory of the unit and three gates: The input gate, the forget gate f_t, and the output gate O_t. At the time stamp t, the LSTM gate receives an input x_t and the previous hidden state h_{t-1}. These inputs are used to compute the activations of the gates and update the cell state and hidden state for the current time step.

Graves (2012a) proposed equations for Long Short-Term Memory (LSTM) networks that describe the operations of the gates and memory cell updates that define LSTMs as follows:

$$f_t = \sigma\left(w_f . [h_{t-1}, x_t]\right) + b_f \quad (3)$$

$$i_t = \sigma\,(w_i . [h_{t-1}, x_t) + b_i \quad (4)$$

$$\overline{c}_t = tanh(w_c . [h_{t-1}, x_t]) + b_c \quad (5)$$

$$c_t = f_t \odot c_t + i_t \odot \overline{c}_t \quad (6)$$

$$o_t = \sigma\,(w_o . [h_{t-1}, x_t) + b_o \quad (7)$$

$$h_t = o_t \odot tanh(c_t) \quad (8)$$

Equation (3) represents the forget gate vector at the time stamp *t*, σ is the activation function, w_f is the weight matrix, $h_{t-1}, x_{t,}\ b_f$ are the hidden state for the previous time stamp, input for the current time stamp and bias for the forget gate respectively. Eq. (4) represents the input vector, it is the input vector, and b_i is the input bias. Equation (5) represents the candidate cell state, *~C_t* is the cell state vector at the time stamp *t, and W_c* is the weight matrix for the candidate cell state. Equations (6-8) represent the cell state update, output gate, and hidden state respectively.

Results and Discussion

Due to the unavailability of a standard dataset for the five categories of OHPE, the dataset has been manually prepared. The model evaluation is conducted at both the expression level and symbol level. For the expression level recognition process, the expressions are encoded with the ground truth values specified in Table (2). The expression level confusion matrix using the ground truths is shown in Table (3). The classification report is shown in Table (4), which includes precision, recall, and F1-score. The lambda layer is applied to the CTC loss function to align the predictions with the actual class labels.

The recognized output y, consists of the concatenation of five sequences, each sequence comprising 100 OHPE expressions. Each expression is treated as an element, with the sequence increment by 1 for every step. Ytrue is modeled as in Eq. (8):

$$Y\_true\ [i] = [i/100] \quad (8)$$

Table (4) represents the precision, recall, f1-score, and support.

Table 3: Expression level confusion matrix

Labels	EF	ME	IDC	PV	MI
EF	96	2	1	1	0
ME	2	95	1	1	1
IDC	1	2	94	2	1
PV	1	1	1	96	1
MI	0	1	1	2	96

Table 4: Performance evaluation at expression level

Expression categories	Precision	Recall	F-score	Support
EF	0.95	0.96	0.95	100
ME	0.94	0.95	0.95	100
IDC	0.95	0.94	0.95	100
PV	0.95	0.94	0.95	100
MI	0.96	0.96	0.96	100
Accuracy	0.96	0.96	96.0	500
Macro Avg	0.95	0.96	0.95	500
Weighted Avg	0.95	0.96	0.95	500

The Ypred has been computed as below in Eq. (9):

$$y_{pred}[i] = \begin{cases} 0 \; if \; 0 \; \leq i < 96 \\ 1 \; if \; 96 \; \leq i < 98 \\ 2 \; if \; 98 \leq i < 99 \\ 3 \; if \; 99 \; \leq i < 100 \\ 0 \; if \; 100 \; \leq i < 102 \\ 1 \; if \; 102 \; \leq i < 197 \\ 2 \; if \; 197 \; \leq i < 198 \\ 3 \; if \; 198 \; \leq i < 199 \\ 4 \; if \; 199 \; \leq i < 200 \\ 0 \; if \; 200 \; \leq i < 201 \\ 1 \; if \; 201 \; \leq i < 203 \\ 2 \; if \; 203 \; \leq i < 297 \\ 3 \; if \; 297 \; \leq i < 299 \\ 4 \; if \; 299 \; \leq i < 300 \\ 0 \; if \; 300 \; \leq i < 301 \\ 1 \; if \; 301 \; \leq i < 302 \\ 2 \; if \; 302 \; \leq i < 303 \\ 3 \; if \; 303 \; \leq i < 399 \\ 4 \; if \; 399 \; \leq i < 400 \\ 1 \; if \; 400 \; \leq i < 401 \\ 2 \; if \; 401 \; \leq i < 402 \\ 3 \; if \; 402 \; \leq i < 304 \\ 4 \; if \; 404 \; \leq i < 500 \end{cases} \qquad (9)$$

The recognition process has been conducted at the symbol level, identifying 25 unique symbols and their total occurrences across 500 OHPEs. Table (5) details the 25 symbols used in physics expressions, along with their class labels and frequencies. The total count of 3900 symbols used for experimentation is presented in Table (5).

Table 5: Symbol class labels and frequency

Class label	Symbol	Frequency	Total symbols
1	Φ	1	100
2	*E*	2	200
3	=	4	400
4	∫	1	100
5	→	5	500
6		1	100
7	d	1	100
8	A	1	100
9	*λ*	1	100
10	m	2	200
11	2	2	200
12	L	2	200
13	÷	2	200
14	I	1	100
15	*μ*	2	200
16	0	2	200
17	*l*	1	100
18	N	1	100
19	*r*	1	100
20	X	1	100
21	(	1	100
22	)	1	100
23	B	1	100
24	s	1	100
25	1	1	100
Total			3900

Figure (5) illustrates the classification accuracy achieved. The performance measure of CRNN-LSTM at the symbol level is carried out by computing precision, recall, F1 score, and the confusion matrix. The confusion matrix provides the counts for TP, FP, FN, and TN. Precision is calculated as the ratio of true positive predictions to the total predicted positives for each of the 25 class labels. Support indicates the actual frequency of each of the 25 symbols, operators, and alphabets used in OHPE across five categories.

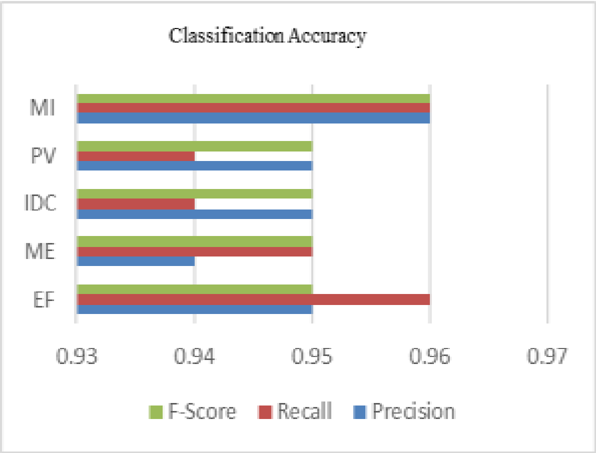


Fig. 5: F-score, recall, and precision for 5 categories of OHPE at expression level

Table 6: Performance evaluation for recognition of symbols

Symbol	Support	Precision	Recall	F1 score
Φ	100	0.90	0.88	0.89
E	200	0.99	0.99	0.99
=	400	0.99	0.99	0.99
∫	100	0.9	0.88	0.89
→	500	0.99	0.99	0.99
	100	0.98	0.98	0.98
d	100	0.98	0.98	0.98
A	100	0.98	0.98	0.98
λ	100	0.98	0.98	0.98
m	200	0.99	0.99	0.99
2	200	0.99	0.99	0.99
L	200	0.99	0.99	0.99
÷	200	0.99	0.99	0.99
I	100	0.9	0.88	0.89
μ	200	0.99	0.99	0.99
0	200	0.99	0.99	0.99
l	100	0.9	0.88	0.89
N	100	0.98	0.98	0.98
r	100	0.9	0.88	0.89
X	100	0.9	0.88	0.89
(	100	0.98	0.98	0.98
)	100	0.98	0.98	0.98
B	100	0.98	0.98	0.98
s	100	0.98	0.98	0.98
1	100	0.98	0.98	0.98

The performance metrics presented in Table (6) highlight the high strength and accuracy of the recognition model across various symbols identified from five categories of physics expressions. The symbol "E" achieved the highest performance, with precision, recall, and F1 scores of 0.99 over 200 samples. The most ambiguous symbols "=", "→", "m," "2", "L" and "÷" are correctly classified, showcasing 0.99 precision. In contrast, symbols with lower support, such as "Φ" and "∫," recorded precision and recall scores of 0.9, leading to F1 scores of 0.89. Overall, most symbols achieved scores of 0.98 or higher, indicating robust performance. Symbols like "d," "A" and "μ" consistently scored 0.98 across 100-200 samples. While symbols like "I," "r," "X," and "l" showed slightly lower performance with precision and recall of 0.9, they still managed F1 scores of 0.89. Parentheses "("and")" performed well, achieving scores of 0.98. This overall strong performance underscores the model's effectiveness in recognizing symbols, particularly for frequently occurring characters in physics expressions.

Table (7) represents the confusion matrix to demonstrate the ambiguity of the symbols with OHPEs. Ambiguous symbols are "Phi," "integration," "I," "small L," "r," "X," and "o", giving them lower scores. The overall accuracy is the weighted average of the precision and recall across all the classes, weighted by the support which is nothing but the number of instances for each class with an overall recognition accuracy of 98.10%.

Table 7: Confusion matrix-symbol level

	Φ	E	=	∫	→		d	A	λ	m	2	L	÷	I	μ	0	l	N	r	X	(	)	B	s	1
Φ	98	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
E	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0
=	0	0	198	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
∫	0	0	0	396	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
→	0	0	2	0	88	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	495	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0
d	0	0	0	0	0	0	98	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	0	0	0	0	98	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
λ	0	0	0	0	0	0	0	0	98	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
m	0	0	0	0	0	0	0	0	0	98	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	198	0	0	0	0	0	0	0	0	0	0	0	0	2	0
L	0	0	0	0	0	0	0	0	0	0	0	198	0	0	0	0	0	2	0	0	0	0	0	0	0
÷	0	0	0	2	0	0	0	0	0	0	0	0	198	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	2	0	198	0	0	0	0	0	0	0	0	0	0	0
μ	0	0	0	0	0	0	0	0	0	12	0	0	0	0	88	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	198	0	0	0	0	0	0	0	0	0
l	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	198	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	88	0	0	0	0	0	0	0
r	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	98	0	0	0	0	0	0
X	0	0	0	0	0	0	0	0	10	0	0	0	0	0	2	0	0	0	0	88	0	0	0	0	0
(	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	88	0	0	0	5
)	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	98	0	0	1
B	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	98	0	0
s	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	98	0
1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	98

Table 8: Comparative study of our proposed method

Author	Model	Dataset	Expression level accuracy	Symbol level accuracy
Zhang *et al*. (2020)	LSTM-CNN	MEs	92%	95%
Graves and Schmidhuber (2009)	Multidimensional RNN	IAM database	N/A	95%
Álvaro *et al*. (2014)	HMM-based	CROHME 2014	90%	90%
MacLean and Labahn (2013)	Relational grammars	Custom HME	85%	85%
Geetha *et al*. (2021)	H2TR model	IAM, RIMES	98.14%	99.35%
Safarzadeh and Jafarzadeh (2020)	Persian words CNN and RNN	Persian and Arabic datasets including IFN/ENIT	99%	Char accuracy: 99.35% digit 99.68%
Bastas *et al*. (2020)	DCNN and RNN	Manual	97.50%	NA
Nurseitov *et al*. (2020)	CNN and MLP, SimpleHTR model	42 types of Cyrillic words contain country names	55.3% for CNN, 57.1% for SimpleHTR	80%
Albahli *et al*. (2021)	Faster-RCNN	Standard MNIST database	99.78%	

Table 8: Continue

Saini *et al*. (2023)	RNN and multidimensional LSTM	Words written by 175 writers	98.67%	90.90%
Hemanth *et al*. (2021)	CNN, RNN and CTC based model	IAM word image dataset	98%	89.38%
Zhan *et al*. (2021)	RNN-CTC	ORAND-CAR-A, ORAND-CAR-B and CVL HDS	Accu. PhPAIS dataset 94.26%	99.43%
Our proposed approach	CRNN-LSTM	500 OHPEs (3900 symbols)	96.10%	98.10%

Recognition of OHPEs is a less explored area, due to limitations of the existing literature, the comparative study has been conducted including other handwriting as well in Table (8) which shows our proposed CRNN-LSTM model has acceptable performance as compared to other proposed methods in the existing literature.

The proposed CRNN-LSTM model outperforms the existing established models with the custom dataset of 500 OHPEs and with 3900 symbols which showcases the robustness and versatility of the proposed model. The proposed CRNN-LSTM model in this study is a combination of high accuracy and adaptability to diverse datasets which includes different categories of physics expressions.

Conclusion

The recognition of OHPE using CRNN-LSTM is conducted at both expression level and symbol level using a dataset of 500 expressions under 5 categories consisting of a total of 3900 mathematical Physics symbols, operators, and alphabets. Overall results demonstrate the high-level accuracy and performance of the proposed CRNN-LSTM models in recognition of OHPE. Recognition accuracy at the expression level is 96.10% across 5 categories and each category of OHPE shows high precision, recall, and F1 Score values of 0.95, which is a balanced performance across different OHPE categories. There is minor misclassification, with each expression category, these errors are minimal and distributed uniformly indicating no significant bias towards any particular category. At the symbol level, the model performance is evaluated on 25 unique symbols that are frequently used in the physics expressions. The proposed model demonstrates consistently high precision, recall, and f1 score across all 25 symbols, with most values are in between 0.97 to 0.99. This indicates the model's robustness in accurately identifying individual symbols. In this, the evaluation of the CRNN-LSTM model shows effectiveness and robustness in the recognition of OHPE at the expression level and symbol level which makes it valuable for both education and scientific purposes. This study demonstrates recognition accuracy at the symbol level is 98.10% and at the expression level is 96.36%. Future work may focus on extending the dataset to consist of more physics expression categories. Future work may also focus on providing solutions to the recognition of ambiguous symbols.

Acknowledgment

Researchers thank Dr. Vishwanath Karad MIT World Peace University, Pune for providing support and guidance.

Funding Information

The authors currently at the submission did not receive any monetary assistance for the investigation, composition, and/or publication of this manuscript.

Author's Contributions

Ujwala Pritam Kolte: Contributed to the literature review, methodology, experimentation, and result analysis.

Sachin Arun Naik: Contributed to the conceptualization, supervision and conclusion, result evaluation.

Ethics

I undersigned that this article has not been published elsewhere. The authors declare no conflict of interest.

References

Albahli, S., Nawaz, M., Javed, A., & Irtaza, A. (2021). An Improved Faster-RCNN Model for Handwritten Character Recognition. *Arabian Journal for Science and Engineering*, *46*(9), 8509–8523. https://doi.org/10.1007/s13369-021-05471-4

Álvaro, F., Sánchez, J.-A., & Benedí, J.-M. (2014). Recognition of On-Line Handwritten Mathematical Expressions Using 2D Stochastic Context-Free Grammars and Hidden Markov Models. *Pattern Recognition Letters*, *35*, 58–67. https://doi.org/10.1016/j.patrec.2012.09.023

Bastas, G., Kritsis, K., & Katsouros, V. (2020). Air-Writing Recognition Using Deep Convolutional and Recurrent Neural Network Architectures. *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 7–12. https://doi.org/10.1109/icfhr2020.2020.00013

Bunke, H., Bengio, S., & Vinciarelli, A. (2004). Offline Recognition of Unconstrained Handwritten Texts Using HMMs and Statistical Language Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(6), 709–720. https://doi.org/10.1109/tpami.2004.14

Chammas, E., & Mokbel, C. (2021). Fine-tuning Handwriting Recognition systems with Temporal Dropout. *ArXiv*, *1*, arXiv:2102.00511.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/bf00994018

Diem, M., Fiel, S., Garz, A., Keglevic, M., Kleber, F., & Sablatnig, R. (2013). ICDAR 2013 Competition on Handwritten Digit Recognition (HDRC 2013). *2013 12th International Conference on Document Analysis and Recognition*, 1422–1427. https://doi.org/10.1109/icdar.2013.287

Dietterich, T. G. (2000). *Ensemble Methods in Machine Learning* (J. Kittler & F. Roli, Eds.; Vol. 1857). Springer International Publishing. https://doi.org/https://doi.org/10.1007/3-540-45014-9_1

Geetha, R., Thilagam, T., & Padmavathy, T. (2021). Retracted Article: Effective Offline Handwritten Text Recognition Model Based on a Sequence-to-Sequence Approach with CNN–RNN Networks. *Neural Computing and Applications*, *33*(17), 10923–10934. https://doi.org/10.1007/s00521-020-05556-5

Graves, A. (2012a). Long Short-Term Memory. In *Supervised Sequence Labelling with Recurrent Neural Networks* (Vol. 385, pp. 37–45). Springer International Publishing. https://doi.org/10.1007/978-3-642-24797-2_4

Graves, A. (2012b). *Supervised Sequence Labelling with Recurrent Neural Networks* (1st ed., Vol. 385). Springer. https://doi.org/10.1007/978-3-642-24797-2

Graves, A., & Schmidhuber, J. (2009). Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks. *Advances in Neural Information Processing Systems*, *21*, 545–552.

Graves, A., Liwicki, M., Fernandez, S., Bertolami, R., Bunke, H., & Schmidhuber, J. (2009). A Novel Connectionist System for Unconstrained Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(5), 855–868. https://doi.org/10.1109/tpami.2008.137

Hemanth, G. R., Jayasree, M., Keerthi, V. S., Akshaya, P., & Saranya, R. (2021). Cnn-Rnn Based Handwritten Text Recognition. *ICTACT Journal on Soft Computing*, *12*(1), 2457–2463. https://doi.org/10.21917/ijsc.2021.0351

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Jain, T., Sharma, R., & Malhotra, R. (2021). Handwriting Recognition for Medical Prescriptions Using a CNN-Bi-LSTM Model. *2021 6th International Conference for Convergence in Technology (I2CT)*, 1–4. https://doi.org/10.1109/i2ct51068.2021.9418153

Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., Bigorda, L. G. i, Mestre, S. R., Mas, J., Mota, D. F., Almazan, J. A., & de las Heras, L. P. (2013). ICDAR 2013 Robust Reading Competition. *2013 12th International Conference on Document Analysis and Recognition*, 1484–1493. https://doi.org/10.1109/icdar.2013.221

Kolte, U., Naik, S., & Kumbhar, V. (2024). A CNN-KNN Based Recognition of Online Handwritten Symbols within Physics Expressions Using Contour-Based Bounding Box (CBBS) Segmentation Technique. *Journal of Computer Science*, *20*(7), 783–792. https://doi.org/10.3844/jcssp.2024.783.792

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791

Ma, J. (2023). A Study on a Hybrid CNN-RNN Model for Handwritten Recognition Based on Deep Learning. *Proceedings of the 1st International Conference on Data Analysis and Machine Learning*, 268–273. https://doi.org/10.5220/0012801000003885

MacLean, S., & Labahn, G. (2013). A New Approach for Recognizing Handwritten Mathematics Using Relational Grammars and Fuzzy Sets. *International Journal on Document Analysis and Recognition (IJDAR)*, *16*(2), 139–163. https://doi.org/10.1007/s10032-012-0184-x

Nurseitov, D., Bostanbekov, K., Kanatov1, M., Alimova, A., Abdallah, A., & Abdimanap, G. (2020). Classification of Handwritten Names of Cities and Handwritten Text Recognition using Various Deep Learning Models. *Advances in Science, Technology and Engineering Systems Journal*, *5*(5), 934–943. https://doi.org/10.25046/aj0505114

Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, *77*(2), 257–286. https://doi.org/10.1109/5.18626

Safarzadeh, V. M., & Jafarzadeh, P. (2020). Offline Persian Handwriting Recognition with CNN and RNN-CTC. *2020 25th International Computer Conference, Computer Society of Iran (CSICC)*, 1–10. https://doi.org/10.1109/csicc49403.2020.9050073

Saini, D. J. B., Kumar, S., Joshi, K., Pathak, A. K., Jain, S., & Singh, A. (2023). A Novel Approach of Image Caption Generator Using Deep Learning. *2023 Third International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS)*, 24–29. https://doi.org/10.1109/icuis60567.2023.00012

Sanap, A., Lomte, V. M., Joshi, A., Nasare, A., & Chandratre, S. (2023). Handwritten Character Recognition from Physical Documentation. *IJRAR-International Journal of Research and Analytical Reviews (IJRAR*, *10*(2), 499–505.

Zhan, H., Lyu, S., Lu, Y., & Pal, U. (2021). DenseNet-CTC: An End-to-End RNN-Free Architecture for Context-Free String Recognition. *Computer Vision and Image Understanding*, *204*, 103168. https://doi.org/10.1016/j.cviu.2021.103168

Zhang, X., Du, J., & Dai, L. (2020). Recognition of Handwritten Mathematical Expressions via a Novel LSTM-CNN Model. *Journal of Pattern Recognition Research*, *15*(1), 33–48.