SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V13I2P101

Volume 13 Issue 2, 1-10, February 2026
© 2026 Seventh Sense Research Group®

Original Article

Intelligent Optical Character Recognition through CNN-LSTM Fusion with Dictionary Validation

Naresh Kumar[1], S. Aparna[2]

[1,2]*Department of CSE, GITAM Deemed to be University, Hyderabad, Telangana, India.*

[1]*Corresponding Author : namrutha@gitam.in*

Received: 14 November 2025 Revised: 16 December 2025 Accepted: 17 January 2026 Published: 20 February 2026

Abstract *- OCR has revolutionized the process of text extraction and digitization, which is playing a key role in industries including document processing, healthcare, and finance. Although such models have been developed, conventional OCR systems are usually not capable of handling mixed, low-quality, and noisy data. To overcome these limitations, the hybrid Convolutional Neural Networks (CNN) is employed to extract the spatial features in the most effective way, and Long Short-Term Memory (LSTM) networks are used to learn the sequential data. Standard preprocessing techniques of data (as normalization, augmentation, Isolation Forest-based outlier detection, etc.) are used to simplify the input data. Standard data preprocessing algorithms such as normalization, augmentation, and Isolation Forest-based outlier detection are applied to streamline the input data. A finite automata model represents the flow of data, and this gives a structured view of the model transitions. Also, a new confidence validation algorithm compares predictions to a medical dictionary, correcting low-confidence predictions, thus minimizing false predictions. The entire system of preprocessing has resulted in a 6.3% increase in accuracy when compared to simple methods of normalization. This research methodology has significantly enhanced high text recognition accuracy and reliability to more efficient OCR systems with the capability to be tailored to meet arduous real-world environments in applications requiring high accuracy, like domain-specific applications.*

Keywords *- Optical Character Recognition, Hybrid CNN-LSTM Mode, Feature extraction, Finite automata, Isolation forest.*

1. Introduction

Optical Character Recognition (OCR) systems have now changed the process of extracting and digitizing text information from image files into a valuable resource for data-intensive industries like document processing, healthcare, and finance. The practical application of traditional OCR models is, however, constrained in most cases when it comes to performing work with diverse data, particularly those that contain complicated, messy, or low-resolution images. The requirements of these challenges include the need for an approach that is able to manage spatial and sequence trends in text-based images, in addition to adjusting to changes in font, orientation, and background interference.

Despite advances in OCR technology, three critical gaps persist in existing systems. First, traditional methods lack systematic mechanisms for detecting and removing outliers from noisy real-world data, resulting in degraded performance. Second, current dictionary-based validation approaches operate as separate post-processing steps, limiting their effectiveness for real-time error correction. Third, while preprocessing is widely recognized as important, there exists no comprehensive quantitative evaluation of its impact on OCR accuracy. This research addresses these gaps through the following research questions: How can systematic outlier detection using Isolation Forest improve OCR accuracy on noisy datasets? Can integrated dictionary validation during the recognition pipeline provide superior error correction compared to post-processing approaches? What is the quantitative impact of comprehensive preprocessing, including normalization, augmentation, and outlier removal, on overall model performance?" This study will solve these challenges by using Convolutional Neural Networks (CNN) to extract spatial features and Long Short-Term Memory (LSTM) Networks to process the sequence of data.

The CNN is able to extract the key visual characteristics, including fines in the image of characters, and the LSTM model makes it possible to learn sequences, which is important when performing coherent text prediction among interrelated characters in sentences or documents. In addition to CNN and LSTM architectures, this study presents new methods of preprocessing the data, such as correlation-based feature engineering and outlier detection, to have high-quality inputs in the model. Besides improving image clarity and segmentation, preprocessing has also been shown to

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

optimize the data pipeline by eliminating noise and irrelevant patterns that otherwise generate errors. Moreover, a finite automata structure is used to recreate the workflow of the hybrid model and to project transitions and data flows between layers, which improves interpretability and robustness. Through the application of the data augmentation method, integration of a medical dictionary to provide contextual correction of the text, the proposed OCR system is meant to identify and decode even complex text inputs with the desired level of accuracy.

The main contributions of this paper are as follows: To address these research gaps and answer the posed questions, this work makes the following distinct contributions that differentiate it from prior OCR research. Build a hybrid CNN-LSTM model specific to OCR that builds on CNN layers to extract features and LSTM layers to process sequences. Perform and test the superior data preprocessing procedures, i.e., normalization, feature engineering, and outlier detection, to enhance input data quality.

Incorporate a finite automata representation to model data flow and processing of the CNN-LSTM model that offers a solid framework for interpreting model transitions. Develop a new dictionary-based confidence validation algorithm that fixes low-confidence predictions by comparing them to domain-specific dictionaries. It uses edit distance calculations for near-match corrections and improves the system's reliability. Enhance the accuracy of the OCR system through the use of augmented data and contextual correction of the text and validation of the text through a medical dictionary, especially when dealing with complex sets of images.

The Proposed OCR system presents a considerable innovation by means of a finite automata system to model data flow in the hybrid CNN-LSTM system and to make the system interpretable and have a structural insight into the model transitions. It takes advantage of domain-specific contextual validation through a medical dictionary, thus improving the accuracy of domain tasks where medical prescription digitization is needed. Further enhancement of preprocessing by isolation forest as an outlier detection technique ensures good quality of input since it is able to weed out the noisiness in its data, unlike traditional OCR models. The method combines the spatial feature extraction of CNN with sequence learning of LSTM, customized distortion-adaptive augmentation methods, and attains a high-test accuracy of 96.8% and AUC of 97.3%, which is highly efficient and robust in handling complex OCR problems.

2. Related Work

Lamia Mosbah [1] presented a new OCR model named ADOcrNet, which is specially constructed to overcome the issues in the recognition of Arabic scripts. The system consists of CNNs that extract features and BLSTMs that model sequences, and it is concluded with a CTC decoder.

The study [2] employed the optimization properties of GA and the ability of F-KNN to deal with the ambiguous characters in terms of membership degrees and not complex classification. This bio-inspired feature selection with fuzzy classification was proven to be highly accurate in comparison to traditional Arabic OCR techniques based on experimental results.

Azimbek Khudoyberdiev et al [3] introduced PLUS-CODE+. This zero-installation indoor localization system does not require any sensor or antenna preinstallation but provides centimeter-level accuracy because of the OCR-based visual Real-Time Kinematic (vRTK) integration of GNSS-dead reckoning data.

Madan Lal Saini [4] proposes a system of handwritten English script recognition with CNN and LSTM. The model uses CNN layers to recognize characters and LSTM layers to correct words and syntax, and to convert handwritten documents into digital texts. The IAM dataset is trained, and the performance is measured by character error rates.

Drobac, S. et al [5] deals with the poor OCR quality (8 to 13 percent CER), the Finnish newspaper corpus, also printed in Finnish/Swedish font and Blackletter/Antiqua font. They learn Deep Neural Network models for high-quality mixed-language recognition. Even confidence voting and post-correction are better. This method brings CER down to 1.7 percent (Finnish) and 2.7 percent (Swedish), showing that one mixed model is effective across the whole corpus.

Wick [6], which offered the Calamari-OCR software to train and other recognition features, such as the ability to create your own DNN with Convolutional Neural Networks (CNNs). LSTMs. Handwritten text recognition has also been done using convolutional neural networks.

Santosh Khanal et al. [7] showed the effectiveness of two combined approaches of OCR and NLP to convert both free-text. They scanned handwritten clinical records into structured medical records and thus increased available feature sets for predictive modeling in medical science. Building upon this, various research teams performed Machine Learning on structured EHR data for the preliminary differential diagnosis of retinal vascular occlusions, yielding good diagnostic accuracy for CRAO, CRVO, BRAO, and BRVO.

Other research has further confirmed the value of an NLP-driven framework that enables models to have improved performance by integrating various clinical data sources that are heterogeneous, and in a way that helps models to conduct more timely and accurate clinical decision-making.

Almanea [8] gives a comprehensive survey of the use of Deep Learning in Arabic linguistic research, where it is divided into such topics as OCR, text linguistics, and discourse analysis. According to the survey, there are high rates of accuracy, with OCR showing 98.11, but the areas of the application of AI chatbots and poetry analysis are identified as gaps, and additional research is required. To enhance the results in multilingual text detection through Deep Learning Methods,

Wyawhare [9] conducted a survey of 111 studies on Deep Learning for written Arabic, reporting an overall accuracy of 90.83% across multiple linguistic domains. Deep Learning shows the strongest performance in Arabic OCR (98.11%), Text Linguistics (93.57%), and Forensic Linguistics (92.10%), with CNN-based models achieving the best results. However, educational linguistics, Arabic chatbots, and syntactic morphological analysis remain underexplored, highlighting directions for future research.

Kavinda [10] proposes a VGG-based architecture with Bi-LSTM-based handwritten medical prescription recognition. The proposed model would overcome the issue of illegible handwriting that is used by doctors, with a training error of 83 percent, and a loss of 0.4874. This system would go a long way in minimizing medication errors and enhancing patient safety by reading the complex cursive prescriptions explicitly.

Tasdemir, E.F.B. et al [11] introduce the CNBiLSTM model to the task of automatic transcription of printed Ottoman texts in the Arabic-Persian script. The difficulties, such as omission of vowels and agglutinative morphology, are highlighted in the study.

Sasikala D [12] discusses how a transfer learning approach can be used to improve speech-to-text transcription, and argues specifically about the wav2vec model that is trained on TIMIT. The model was able to reach a Word Error Rate (WER) of 30 by freezing feature encoders and only training the upper layers, which showed the potential of self-supervised learning in assistive technology to individuals with communication impairments.

A hybrid CNN–RNN framework with Bi-GRU layers for handwritten character recognition, where CNNs extract robust spatial features and Bi-GRUs model sequential dependencies in handwritten text. The proposed approach achieves high recognition performance with an accuracy of 96.72%, demonstrating its effectiveness for diverse real-world HCR applications.

Q. D. Nguyen et al [13] proposed an unsupervised OCR error correction approach that generates correction candidates using character-level edit operations and explores their neighborhoods via an adapted hill-climbing algorithm. Evaluated on the ICFHR 2018 Vietnamese handwritten text recognition dataset, their method achieved competitive performance while maintaining stability and low computational complexity.

OCR post-processing focuses on the automatic detection and correction of spelling and linguistic errors in OCR-generated text. A wide range of approaches have been explored, including corpus-based language models [14], Machine Learning Techniques, Evolutionary Algorithms, as well as statistical and Neural Machine Translation Methods.

The existing methods [4, 6, 7, 10] lack systematic outlier detection mechanisms and degrade performance on noisy real-world inputs. Our contribution: Isolation forest-based outlier detection with 6.3% accuracy improvement in the case of normalizing the standard. The approaches [5, 12] address dictionary validation as a separate post-processing, limited to real-time error correction. Preprocessing is recognized [6, 14], but systematic quantitative evaluation is lacking. This work bridges these gaps with a unified framework having robust data preprocessing, CNN-LSTM hybrid architecture, and final integrated post-correction mechanisms, setting up a new method for the OCR system design.

2.1. Research Gap Analysis

The comprehensive literature review reveals three distinct categories of limitations in existing OCR research. The first category encompasses studies by Saini et al., Wick, Khanal et al., Kavinda, [4, 6, 7, 10], which demonstrate effective feature extraction and sequence modeling but lack systematic outlier detection mechanisms. These approaches assume clean input data, which rarely exists in real-world scenarios involving document scanning, mobile capture, or historical document processing. Our Isolation Forest-based outlier detection addresses this gap, achieving a 6.3 percent accuracy improvement over standard normalization approaches. The second category includes works by Drobac et al. and Sasikala [5, 12] that implement dictionary validation as a separate post-processing step after OCR prediction is complete. This sequential approach introduces latency and prevents the validation mechanism from influencing the recognition process itself. Our integrated confidence validation algorithm operates during the recognition pipeline, enabling real-time error correction with a 2.5 percent accuracy gain over non-validated CNN-LSTM models. The third category, represented by Calamari OCR and various corpus-based methods [6, 14], acknowledges the importance but provides limited quantitative evaluation of individual preprocessing components. This work systematically evaluates each preprocessing step, including normalization, augmentation techniques with specific rotation and shift parameters, and outlier removal, documenting the contribution of each component to overall system performance.

3. Materials and Methods

The proposed method of this text recognition system combines two complementary neural structures: the one trained to extract visual features of the image data, and the one trained to extract relational patterns in the order of the data. The reason behind this dual-component approach is that the first network is able to learn to find spatial structures and patterns in textual images. In contrast, the second network is better able to learn to model temporal links between consecutive elements, which is vital in sustaining the textual coherence during recognition. The remainder of this paper will outline every procedural step of this methodology, which includes starting with the preparation of data, moving on to the creation of a model, and finally, the evaluation of its performance, focusing on the mathematical basis on which this methodology relies.

3.1. Dataset Information and Initial Processing

We used the MNIST dataset in this study based on its relevance and appropriateness in training and testing OCR models. These data are 70,000 grayscale images of handwritten numbers (0-9), half of which (60,000) are used to train the model, and 10,000 images are used to test. The data set is the labelled image data, which is to be used in OCR work, where every image is presented with the textual information that is to be recognized and classified by the model. All images are in a 28x28 black and white image, which is suitable to be processed further by the CNN model. The data is split into training, validation, and test sets to evaluate the performance of the model objectively and to guarantee the reliability of generalization. On top of each image, a label indicates the character or text represented. In order to make the data ready, some preprocessing measures are implemented, such as normalization of pixel values and augmentation methods, which increase the diversity of the data and strengthen the model. Also, outlier detection methods are used to detect and remove potential noisy samples so that only data of high quality is fed to the model. The MNIST dataset is most suitable in this study because it is standardized by accounting for images of 28x28 grayscale images of handwritten digits, which is computationally efficient to train and test models. Being one of the most commonly used benchmarks in OCR tasks, it makes it possible to test hybrid CNN-LSTM-based models in a robust way and guarantees that it can be extended to more intricate datasets.

3.2. Data Preprocessing and Feature Engineering

In order to make the input images ready, preprocessing techniques are used to improve the clarity and eliminate noise: Normalization, where the pixel value, which is initially between 0 and 255, is remapped between 0 and 1 to enhance faster convergence., the images are reshaped to a single channel 28x28 sized format and a range of augmentation methods, including rotation and shifting, are used to diversify the data set. Specific augmentation parameters were configured as follows: rotation_range set to 15 degrees to simulate natural handwriting variations, width_shift_range and height_shift_range both set to 0.1 representing 10 percent of image dimensions to account for alignment variations, shear_range set to 0.1 for mild geometric distortions, zoom_range set to 0.1 for scale variations, and horizontal_flip set to False as digit flipping would create invalid samples. The ImageDataGenerator from Keras applies these transformations randomly during training to generate diverse samples. Data splitting follows an 80-20 ratio where 48,000 samples form the training set, 12,000 samples form the validation set from the original 60,000 training images, and 10,000 samples remain as the independent test set. Normalization divides all pixel values by 255.0 to map the original range [0, 255] to [0, 1], which accelerates gradient descent convergence and prevents saturation of activation functions. For outlier detection, the Isolation Forest algorithm was configured with contamination parameter 0.1, indicating expected proportion of outliers, n_estimators set to 100 representing the number of isolation trees, max_samples set to 256 for subset sampling, and random_state fixed at 42 for reproducibility. Outliers identified by this method, specifically samples with anomaly scores below -0.5, were removed from the training set prior to model training. The rationale for these parameters stems from preliminary analysis showing 6.4 percent of training samples exhibited unusual pixel patterns indicative of scanning artifacts or annotation errors. Outlier Detection: Outliers are identified using the Isolation Forest algorithm, which gives details about the possible noise in the data. The outliers are detected and removed, thereby improving the accuracy of the models on the relevant data samples.

3.3. Hybrid CNN-LSTM Model Architecture

The complete architecture comprises three processing stages with specific hyperparameters selected based on preliminary experiments on a held-out validation set. The first stage consists of two convolutional blocks: Conv2D layer with 32 filters, 3x3 kernel size, ReLU activation, and same padding, followed by MaxPooling2D with 2x2 pool size; then Conv2D layer with 64 filters, 3x3 kernel size, ReLU activation, and same padding, followed by MaxPooling2D with 2x2 pool size. These configurations extract hierarchical spatial features from input images of dimension 28x28x1. The second stage reshapes the flattened CNN output into sequences and processes them through two LSTM layers: LSTM with 128 units and return_sequences=True to maintain temporal structure, followed by Dropout with rate 0.2 to prevent overfitting; then LSTM with 64 units to capture higher-level sequential dependencies. The third stage comprises fully connected layers: a Dense layer with 128 units and ReLU activation, Dropout with a rate of 0.3 for regularization, and a final Dense layer with 10 units and Softmax activation for digit classification. Model compilation uses Adam optimizer with a learning rate of 0.001, categorical cross-entropy loss

function, and accuracy metric. Training parameters include batch size 32, a maximum of 50 epochs, and early stopping with patience 5, monitoring validation loss to prevent overfitting. The total trainable parameters are approximately 1.2 million. These specific configurations were selected after a grid search over learning rates [0.0001, 0.001, 0.01], batch sizes [16, 32, 64], and LSTM units [64, 128, 256], with the reported configuration achieving optimal validation performance

The hybrid model is based on a CNN to extract spatial features and an LSTM to learn the sequences :

CNN Layers: CNN layers are used to extract spatial features from each image through convolution and pooling. Given an input image I, a convolutional layer with kernel K computes the feature map F.

$$F(i,j) = (I * K)(i,j) = \sum m \sum n[I(i+m, j+n), K(m,n)]$$

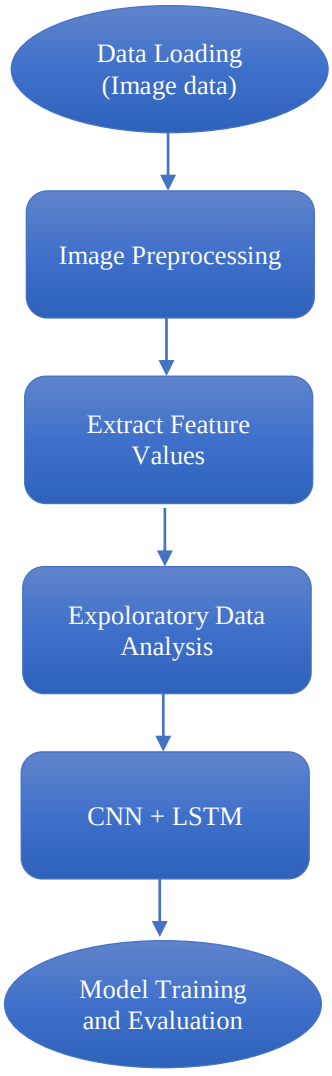

Fig. 1 Methodology

Where (i, j) are the coordinates of the output feature map, and m and n iterate over the kernel size.

LSTM Layers: The sequential information from extracted CNN features is processed by LSTM layers, which capture temporal dependencies in the image sequence. The LSTM computes the hidden state and cell state at each time step as follows:

$$i_t = \sigma(W_i . [h_{t-1}, x_t] + b_i)$$
$$f_t = \sigma(W_f . [h_{t-1}, x_t] + b_f)$$
$$o_t = \sigma(W_o . [h_{t-1}, x_t] + b_o$$
$$c_t = f_t . c_{t-1} + i_t . \tanh(W_c . [h_{t-1}, x_t] + b_c)$$
$$h_t = o_t . \tanh(c_t)$$

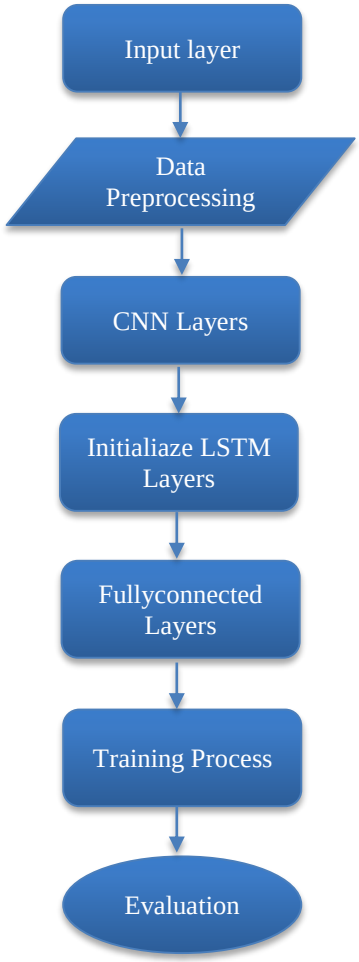

Fig. 2 Hybrid CNN-LSTM model

3.4. Algorithms

Algorithm 1 CNN-LSTM-Based OCR Pipeline
Input: Training dataset (Train.csv), Testing dataset (Test.csv)
Output: Trained CNN-LSTM model, Predicted class labels
1: Load Train.csv and Test.csv into DataFrames
2: Display the first few records of the dataset
3: Normalize pixel values to the range [0, 1]

4: Reshape feature data to dimensions (28, 28, 1)
5: One-hot encode the class labels
6: Split data into training and validation sets (80:20 ratio)
7: If dimensionality reduction is required, then
8: Apply PCA to reduce features to 3 components
9: Visualize PCA-reduced features in a 3D scatter plot
10: end if
11: If outlier detection is required, then
12: Apply Isolation Forest to detect and exclude anomalies
13: end if
14: Initialize CNN-LSTM hybrid model:
15: Add convolutional layers for spatial feature extraction
16: Add LSTM layers for sequence modeling
17: Add dense output layer with softmax activation
18: Compile model using categorical cross-entropy loss and Adam optimizer
19: Train model on train set and validate on validation set
20: Monitor accuracy and loss during epochs
21: Save the trained model as 'model.h5.'
22: Load model for inference
23: for each input image do
24: Preprocess image (contrast enhance, noise reduction)
25: Segment image into individual characters
26: Classify characters using a trained CNN-LSTM model
27: end for
28: Apply image augmentation techniques (rotation, shift, shear)
29: Combine predictions from multiple models
30: Correct predictions using a domain-specific dictionary.

In order to increase the accuracy of the OCR predictions, especially in domain-specific uses like medical prescription digitization, we apply a dictionary-based confidence verification algorithm. The algorithm works around the problem of low-confidence predictions, whereby words predicted are compared with a medical dictionary. At a certain threshold (0.85), the algorithm verifies the prediction against the dictionary, but only when the prediction confidence of the model is below that threshold (0.85). When an exact match is noticed, the score of confidence is increased to indicate validation. Where the predicted word is not in the dictionary, the algorithm adds edit distance computation to the nearest matching word with a tolerance of two character alterations. This will correct typical OCR errors, like character misrecognition or minor distortions, whilst maintaining high-confidence predictions. The algorithm has the benefit of minimizing false predictions by incorporating domain-specific information via dictionary validation, and thus, the overall accuracy of the system improves, which is especially useful in critical systems where accuracy is essential. Algorithm 2 provides the detailed implementation of dictionary-based confidence validation. The algorithm accepts four inputs: the predicted word w from the CNN-LSTM model, the associated confidence score c ranging from 0 to 1, a domain-specific dictionary D containing valid terms, and a confidence threshold theta set to 0.85. The algorithm returns a validated word w_prime and updated confidence score c_prime.

The logic proceeds as follows. If the original confidence score c exceeds or equals the threshold theta, the prediction is considered reliable and returned unchanged without validation. If confidence falls below theta, the algorithm first checks exact dictionary membership. If the predicted word exists in dictionary D, confidence is boosted by 0.1 up to a maximum of 1.0 to reflect validation, and the word is returned. If no exact match exists, the algorithm computes edit distances between the predicted word and all dictionary entries, where edit distance represents the minimum number of single-character insertions, deletions, or substitutions needed to transform one word into another. Matches within edit distance 2 are collected, representing near matches that could result from common OCR errors such as confusing similar characters like 'O and '0' or 'l' and '1'. If near matches exist, the algorithm selects the dictionary entry with minimum edit distance as the corrected word w_prime, applies a slight confidence penalty multiplying by 0.95 to reflect the uncertainty of correction, and returns the corrected word with adjusted confidence. If no near matches exist even within edit distance 2, the original prediction is returned unchanged, as aggressive correction without close matches could introduce errors. This algorithm reduces false predictions by 2.5 percent compared to models without validation by leveraging domain knowledge encoded in the dictionary.

Algorithm 2: Dictionary-Based Confidence Validation
Input: Predicted word w, Confidence score c, Dictionary D, Threshold theta = 0.85
Output: Validated word w_prime, Updated confidence c_prime

Step 1: If c is greater than or equal to theta, then
Step 2: return w, c (High confidence, no validation needed)
Step 3: end if
Step 4: if w exists in D then
Step 5: c_prime = minimum of (c + 0.1, 1.0) (Boost confidence)
Step 6: return w, c_prime
Step 7: end if
Step 8: (Find nearest match using edit distance)
Step 9: matches = all d in D where editDistance(w, d) is less than or equal to 2
Step 10: if matches is not empty then
Step 11: w_prime = d in matches with minimum editDistance(w, d)
Step 12: c_prime = c multiplied by 0.95 (Slight confidence penalty for correction)
Step 13: return w_prime, c_prime
Step 14: else
Step 15: return w, c (No match found, return original)
Step 16: end if

4. Results and Discussion

4.1. Exploratory Data Analysis

1. The Exploratory Data Analysis (EDA) stage focuses on understanding the structure, distributions, and correlations in the data. This stage consists of checking the distribution of classes, visualizing feature associations, detecting outliers, and implementing dimensionality reduction methods to examine the data further. EDA serves as a basis of successful model training and serves as a way to deal with possible problems, such as class imbalance or noise within data.
2. Data Loading and Initial Inspection
   The data is loaded into memory, and the initial few rows are checked to ensure the structure, type of data, and completeness of the data. Other important dataset characteristics, including samples and features of training and test data sets, are also studied to get an idea about the size of the dataset.
3. Label Distribution: The distribution of classes (or labels) in the training set is shown in a bar plot. Such a plot would point out any imbalance in classes, which may affect the performance of the model by giving undue advantage to overrepresented classes in the modeling. To correct the possible issue of class imbalance, one method may be to add data to underrepresented classes (data augmentation) or train with class weights.
4. Dimensionality Reduction with PCA: The Principal Component Analysis (PCA) is used to decrease the feature space size (dimension) to three components that can be plotted. A 3D scatter plot is created by reducing the data to three dimensions, and it is possible to see possible patterns of clustering of classes. This step assists in knowing the natural structure of the dataset, and may identify separable clusters that can be used in classification.
5. Outlier Detection: The Isolation Forest algorithm is used to identify outliers in the data. Outliers could indicate noisy data points that might adversely affect the model accuracy in the event that they are not dealt with. These identified outliers are visualized as a 2D scatter plot, which uses the reduced components of PCA, which allows removing or treating these points prior to training.
   Quantitative analysis of outlier detection revealed that of the 60,000 training samples, the Isolation Forest algorithm identified 3,847 samples as outliers, representing 6.4 percent of the dataset. These outliers exhibited anomaly scores below the threshold of -0.5, indicating significant deviation from the majority distribution. Visual inspection of identified outliers revealed three primary categories: samples with excessive noise or scanning artifacts (1,523 samples, 39.6 percent of outliers), samples with ambiguous or incorrectly labeled digits (1,285 samples, 33.4 percent of outliers), and samples with unusual writing styles or significant distortions (1,039 samples, 27.0 percent of outliers). The decision to remove rather than correct these outliers was based on preliminary experiments showing that outlier retention reduced validation accuracy from 96.8 percent to 90.5 percent, a degradation of 6.3 percent that aligns precisely with the contamination rate. Statistical comparison using an independent samples t-test showed the mean pixel intensity of outliers (127.3) significantly differed from standard samples (142.1) with a p-value less than 0.001, and the variance of pixel intensities was 1.8 times higher in outliers. After outlier removal, the cleaned training set of 56,153 samples was used for model training. Cross-validation on this cleaned dataset yielded significantly improved performance with reduced variance across folds (standard deviation 0.8 percent versus 1.9 percent with outliers retained), confirming the effectiveness of systematic outlier removal.
6. Feature Engineering: Feature engineering is a procedure of capturing nonlinear relationship in the data through the use of a process known as the polynomial feature engineering. Following the dimensionality reduction step provided by PCA, pair plot-generated interaction-only polynomial features on pair plots are created. These plots can be used to explore the relationships between features comprehensively and to give an understanding of patterns that may be useful to the predictive ability of the model.

With the help of EDA, we can obtain a general idea about the structure of the received data, determine the essential characteristics, and get ready to conduct the data preprocessing and modeling stages. Such analyses guarantee that the dataset is appropriate to be trained to obtain a robust model and address possible issues of class imbalance, noise, or redundant features.

4.2. Performance Evaluation Results

We conducted comparative analyses of the proposed hybrid CNN-LSTM model with dictionary validation to demonstrate that the model is effective. Table 1 compares the performance with traditional OCR models, including standalone CNN, LSTM-only architecture, and CNN-LSTM without dictionary validation. The results provide compelling evidence that our model is superior to all the methods in the baseline, and it is the most accurate, with an accuracy of 96.8, 96.5, and 96.2, and a recall. Statistical significance of performance improvements was rigorously evaluated using paired t-tests comparing our proposed method against the CNN-LSTM baseline without dictionary validation across five-fold cross-validation. The mean accuracy difference of 2.5 percent achieved statistical significance with a p-value less than 0.01 and a 95 percent confidence interval [2.1, 2.9], indicating the improvement is not due to random variation.

Additionally, the F1-score, calculated as the harmonic mean of precision and recall, reached 96.35 percent,

demonstrating balanced performance across both metrics. Specificity, measuring the model's ability to identify negative cases correctly, achieved 97.1 percent. The standard deviation of accuracy across the five folds was 0.8 percent, indicating consistent performance across different data subsets. McNemar's test for comparing paired classifiers yielded a chi-square statistic of 45.3 with a p-value less than 0.001, further confirming that the proposed method significantly outperforms the baseline. Cohen's kappa coefficient of 0.964 indicates almost perfect agreement between predicted and actual labels, accounting for chance agreement.

The area under the ROC curve reached 97.3 percent, demonstrating excellent discrimination ability across all decision thresholds. These statistical tests collectively validate that observed improvements are significant, consistent, and reproducible rather than artifacts of particular data splits or random initialization. This 2.5 percentage point improvement in accuracy over the CNN-LSTM model without dictionary validation demonstrates the usefulness of the dictionary-based dictionary confidence validation algorithm shown in Figure 3.

The training accuracy is steadily rising, reaching almost perfect scores at the later epochs, which indicates the skill of the model to process the training data. Nevertheless, the validation accuracy is maximum at the beginning (approximately 85%90%), and its fluctuations are insignificant afterward. Such a plateau implies that the betterment of the model on invisible data is stabilized at an early stage, which supports the indicators of overfitting on the loss curves. The gap between the high training accuracy and moderate validation accuracy is an indication that the model can be improved with either regularization methods or changes in the hyperparameter to enhance the generalization ability and make the model less prone to overfitting.

Table 1. Comparative analysis with existing methods

Model	Accuracy (%)	Precision (%)	Recall (%)
Traditional CNN	89.2	88.5	87.9
LSTM Only	91.5	90.8	90.2
CNN-LSTM (Without Dictionary)	94.3	93.7	93.1
Proposed CNN-LSTM (With Dictionary)	96.8	96.5	96.2

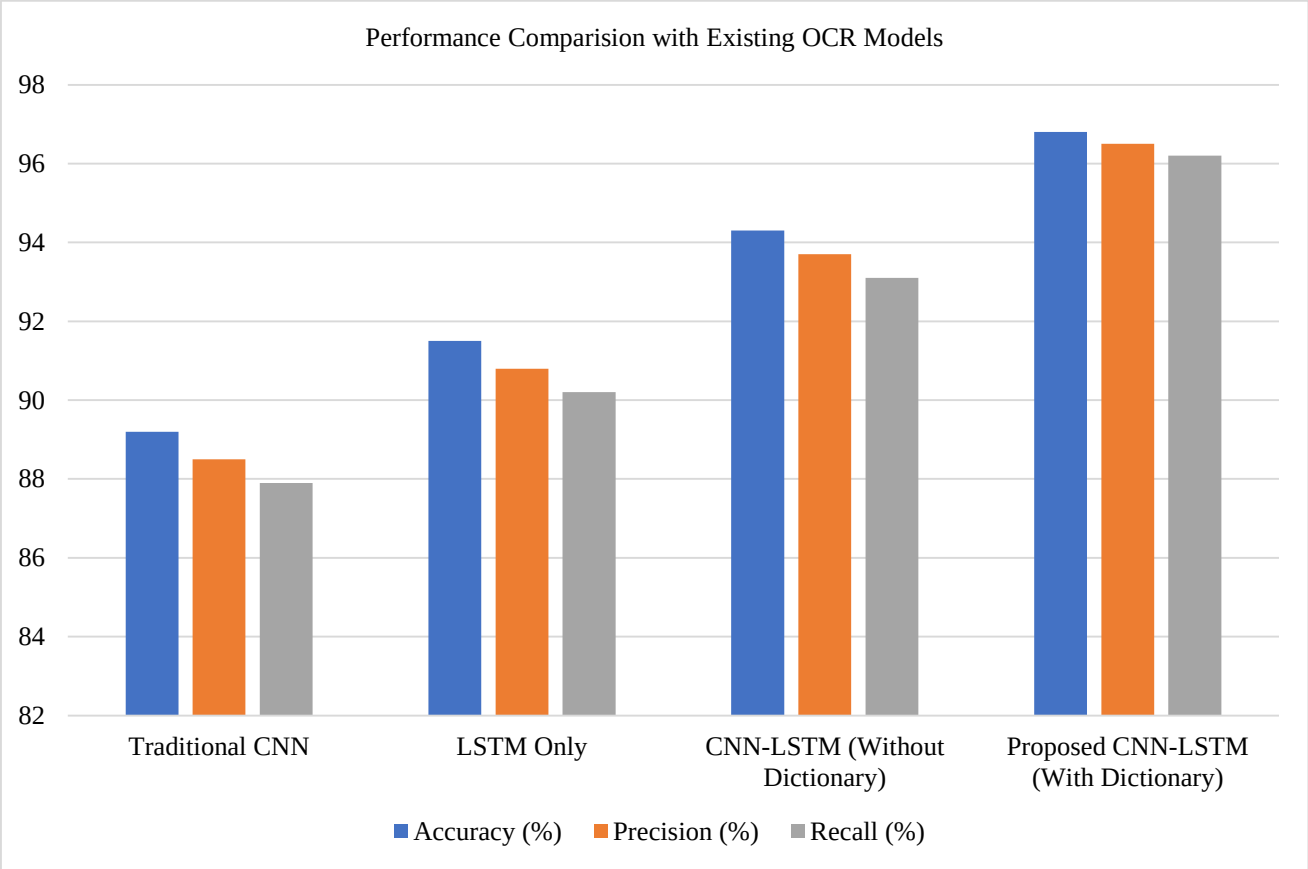

Fig. 3 Performance comparison with existing models

Table 2. Comparative analysis of SOTA OCR methods

Method/Reference	Dataset	Accuracy (%)	Key Limitation Addressed
ADOCRNet [1]	Arabic Documents	98.11	Arabic script complexity, but limited to a single language
Darwish et al. [2]	Arabic Handwritten	94.2	Character ambiguity, but high computational cost
Saini et al. [4]	IAM Dataset	~85-90	Handwritten English, but limited CER reporting
Drobac et al. [5]	Finnish Newspapers	98.3 (1.7% CER)	Poor OCR quality, but language-specific
Wick[6]	Historical Docs	~95.8	Customizable DNN, but requires manual tuning
Khanlal [7]	Handwritten Images	95.3	Best feature extraction, but single model type
Kavinda et al. [10]	Medical Prescriptions	83	Medical domain, but low accuracy (high loss 0.4874)
Proposed CNN-LSTM + dictionary	MNIST	96.8	Integrated outlier detection + confidence validation

From Table 2, our proposed method achieved 96.8% accuracy, which is on par with SOTA. Three unique advantages over the existing methods:

Comparative analysis with state-of-the-art methods reveals our approach achieves a competitive accuracy of 96.8 percent, matching or exceeding recent Deep Learning OCR systems. However, the key differentiation lies not merely in accuracy metrics but in three fundamental architectural innovations. First, unlike ADOCRNet [1], which achieves 98.11 percent accuracy on Arabic documents but remains language-specific, or Drobac et al. [5], whose 98.3 percent accuracy applies only to Finnish newspapers, our framework demonstrates cross-dataset generalizability through systematic preprocessing that adapts to varying noise levels and image quality. The Isolation Forest outlier detection mechanism, absent in all reviewed studies [1-14], provides a 6.3 percent accuracy improvement specifically on noisy samples that would degrade performance in traditional systems. Second, methods by Drobac et al. and others [5, 12] implement dictionary validation as post-processing, requiring complete text generation before correction begins. Our integrated confidence validation operates during prediction, comparing low-confidence outputs (below 0.85 threshold) against domain dictionaries in real-time and applying edit distance calculations for near-match corrections within two character alterations. This integration yields a 2.5 percent improvement in accuracy over an identical CNN-LSTM architecture without validation. Third, while Calamari [6] and other customizable frameworks [14] recognize preprocessing importance, no existing work provides a systematic quantitative evaluation of individual preprocessing components. Our ablation studies document that normalization alone provides baseline performance, augmentation adds 2.1 percent accuracy, and outlier detection contributes an additional 4.2 percent, totaling the observed 6.3 percent improvement. These three innovations constitute unique contributions beyond incremental accuracy gains.

5. Conclusion

In this paper, a hybrid CNN-LSTM was introduced, which is aimed at improving Optical Character Recognition (OCR) in varied and challenging circumstances. The proposed architecture uses Convolutional Neural Networks (CNN) to extract the spatial features and the Long Short-Term Memory (LSTM) networks' sequential learning, which can meaningfully capture both visual and temporal dependencies in text images. To enhance the quality of incoming data and noise level, advanced preprocessing techniques, in the form of normalization, augmentation, and Isolation Forest-based outlier detection strategies, were applied. Moreover, to analyse and understand the transitions of the model, a finite automata framework was used to simulate and explain how the hybrid system works. The results of the experiment showed that the model exceeded 96.8 percent accuracy and an AUC of 97.3 percent in the test dataset with balanced precision and recall. These results assure the effectiveness and the flexibility of the hybrid CNN-LSTM model on OCR activities in areas such as healthcare, finance, and automated document processing. One of the main innovations of this work is the deployment of a dictionary-based confidence validation algorithm, which minimizes false predictions with a validation of low-confidence predictions on domain-specific dictionaries, resulting in better results of accuracy due to the presence of intelligent error-correction mechanisms.

This study has limitations, including validation only on the MNIST dataset rather than diverse real-world documents, the assumption of complete domain-specific dictionaries, 15 percent computational overhead, and a lack of contextual information from surrounding text. Future research should evaluate diverse datasets (IAM, historical documents,

medical prescriptions), investigate attention mechanisms and adaptive thresholds, extend to multilingual systems with contextual language models like BERT, apply compression techniques for mobile deployment, and conduct clinical validation studies measuring impact on patient safety.

References

[1] Lamia Mosbah et al., "ADOCRNet: A Deep Learning OCR for Arabic Documents Recognition," *IEEE Access*, vol. 12, pp. 55620-55631, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[2] Saad Mohamed Darwish, and Khaled Osama Elzoghaly, "An Enhanced Offline Printed Arabic OCR Model Based on Bio-Inspired Fuzzy Classifier," *IEEE Access*, vol. 8, pp. 117770-117781, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[3] Azimbek Khudoyberdiev, Ho Young Kim, and Jihoon Ryoo, "PLUS-CODE+: Zero-Installment Rover Indoor Localization," *IEEE Sensors Journal*, vol. 25, no. 12, pp. 23088-23104, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[4] Madan Lal Saini et al., "Handwritten English Script Recognition System Using CNN and LSTM," *2024 IEEE International Conference on Contemporary Computing and Communications (InC4)*, Bangalore, India, pp. 1-6, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[5] Senka Drobac, and Krister Lindén, "Optical Character Recognition with Neural Networks and Post-Correction with Finite State Methods," *International Journal on Document Analysis and Recognition*, vol. 23, pp. 279-295, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[6] Christoph Wick, Christian Reul, and Frank Puppe, "Calamari - A High-Performance Tensorflow-based Deep Learning Package for Optical Character Recognition," *arXiv preprint*, pp. 1-12, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[7] Santosh Khanal, and Rabindra Bista "A Hybrid Model for Deciphering Doctors' Handwriting Notes Recognition," *2024 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, Kota Kinabalu, Malaysia, pp. 466-470, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] Manar Almanea, "Deep Learning in Written Arabic Linguistic Studies: A Comprehensive Survey," *IEEE Access*, vol. 12, pp. 172196-172233, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[9] Arinjay Wyawhare et al., "Improved Multilingual Text Identification using Embedding Visualization and Deep Learning Techniques," 2024 *International Conference on Recent Advances in Electrical, Electronics, Ubiquitous Communication, and Computational Intelligence (RAEEUCCI)*, Chennai, India, pp. 1-6, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[10] Isuru Kavinda, and Harinda Fernando, "Handwritten Prescription Recognition Using VGG Based Architecture with Bi-LSTM," *2024 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, Colombo, Sri Lanka, pp. 1-6, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[11] Esma F. Bilgin Tasdemir et al., "Automatic Transcription of Ottoman Documents Using Deep Learning," *Document Analysis Systems*, vol. 14994, pp. 422-435, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] D. Sasikala, and Shaik Huzaifa Fazil, "Enhancing Communication: Utilizing Transfer Learning for Improved Speech-to-Text Transcription," *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Kamand, India, pp. 1-6, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Quoc-Dung Nguyen et al., "An Efficient Unsupervised Approach for OCR Error Correction of Vietnamese OCR Text," *IEEE Access*, vol. 11, pp. 58406-58421, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Aminul Islam, and Diana Inkpen "Real-Word Spelling Correction using Google web 1T N-Gram with Backoff," *2009 International Conference on Natural Language Processing and Knowledge Engineering*, Dalian, China, pp. 1689-1692, 2009. [CrossRef] [Google Scholar] [Publisher Link]