A Transformer-Based Optical Character Recognition Framework with Unified Residual Recurrent Networks for Multilingual Handwritten Documents

Dadapeer
Department of Computer Science and Engineering, Ballari Institute of Technology and Management, Ballari, India | Visvesvaraya Technological University, Belagavi, India
dpbitm@gmail.com (corresponding author)

Yeresime Suresh
Department of CSE–Artificial Intelligence, Ballari Institute of Technology and Management, Ballari, India | Visvesvaraya Technological University, Belagavi, India
suresh.vec04@gmail.com

Received: 20 October 2025 | Revised: 17 November 2025 | Accepted: 25 November 2025

Licensed under a CC-BY 4.0 license | Copyright (c) by the authors | *DOI: https://doi.org/10.48084/etasr.15667*

ABSTRACT

Multilingual handwritten Optical Character Recognition (OCR) faces major challenges due to diverse writing styles, script variations, and limited generalization across languages. Existing OCR systems often fail to handle multilingual handwritten data efficiently, resulting in poor segmentation and recognition accuracy. To overcome these challenges, this paper proposes a Transformer-based OCR architecture with a Unified Residual Recurrent Neural Network (TrOCR-URRNN-MLD). The proposed framework integrates Local Sample-Weighted Multiple Kernel Clustering (LSMKC) for effective text segmentation and Adaptive Multi-scale Gaussian Co-occurrence Filtering (AMGCF) for noise suppression and clarity enhancement. A Feature-Affine Residual Network (FA-ResNet) embedded in the Transformer extracts robust spatial-semantic features, while the URRNN with Connectionist Temporal Classification (CTC) efficiently models sequential dependencies. Furthermore, the Secretary Bird Optimization Algorithm (SBOA) optimizes the model parameters for improved performance. Experiments on IAM and Kannada Char74k datasets show that TrOCR-URRNN-MLD surpasses existing OCR models in accuracy, precision, recall, and F1-score.

Keywords-handwritten Optical Character Recognition (OCR); transformer architecture; Unified Residual Recurrent Neural Network (URRNN); multilingual document recognition; feature-affine residual network (FA-ResNet); Secretary Bird Optimization Algorithm (SBOA)

I. INTRODUCTION

OCR is a fundamental task in many applications, with many different approaches that have been proposed. Currently, the most common OCR models rely on LSTM-connectionist temporal classification or sequence-to-sequence attention methods, which take input from a sequence of slices and produce a sequence of recognition outputs [1]. This method enables the development of OCR systems without explicit character segmentation, using character-transition data to automatically learn inter-character correlations. Many modern open systems, such as Tesseract, rely on this method and are currently achieving satisfactory recognition in English text [2]. However, for other script languages, such as mixed Kannada and English, accuracy falls behind, mainly because these languages have more characters than the English language. There are hundreds of character classes in mixed Kannada and English, and even considering pairwise pairings, there are many scenarios that could occur [3]. Hence, in contrast to English, there is a need for explicit character segmentation and individual character recognition for such scripts [4].

The spread of digital content has propelled the demand for effective and precise OCR systems, especially for handwriting in multilingual documents [5]. Handwritten OCR (HOCR) is instrumental in transforming handwritten notes and forms into machine-readable form, thereby accelerating the digitization of historical collections, enhancing document management systems, and promoting accessibility for various linguistic communities [6]. High accuracy rates have proven difficult for traditional OCR techniques to attain, particularly when handling a variety of writing languages and styles [7].

Emerging technologies have transformed this area, making it possible to create reliable models that can extract intricate patterns and features [5]. The goal is to increase the accuracy of recognition in a variety of languages, including—but not limited to—Kannada and English [8]. To improve training and reduce overfitting, data augmentation is used along with addressing the difficulties presented by different handwriting styles and language scripts [9]. HCR-Net [7], a transfer learning-based script-independent framework, achieved high generalization across 40 multilingual datasets, including Bangla, Hindi, Urdu, and Kannada. In [10], a weighted SVM classifier was optimized with the Sine Cosine Algorithm (SCA) for Marathi characters, addressing symbol complexity but suffering from high character error rates. In [11], a DSRNN-MaxEnt classifier was integrated with Nomograph-based IMVO feature extraction for offline handwritten recognition, improving segmentation through Gaussian filtering and projection profiles. In [12], an end-to-end deep learning framework was proposed for historical Ethiopic handwritten text recognition, using a hybrid CTC-attention architecture and achieving a high F1-score but limited accuracy due to dataset constraints. In [13], a systematic review of deep learning methods for historical Arabic manuscripts highlighted advances in feature extraction, sequence modeling, and recognition performance across complex handwritten texts. In [14], Capsule Networks (CapsNet) were employed for recognizing handwritten Devanagari manuscripts, leveraging spatial hierarchies but reporting reduced precision. In [15], an optimized SVM-based framework was presented for Arabic handwritten character recognition, improving classification accuracy while addressing script complexity.

Designing and developing an effective HOCR system for multilingual documents, particularly in Kannada and English, poses significant challenges due to the diversity in scripts, handwriting styles, and document quality. Handwritten characters exhibit a high degree of variability in shape, stroke, and spacing, further complicated by the differences in linguistic structures between the Kannada and English languages. Existing OCR systems struggle with accurately recognizing such complex handwritten texts, leading to errors in character segmentation and recognition. These limitations motivate the proposed Transformer-based OCR with Unified Residual Recurrent Neural Network (TrOCR-URRNN-MLD), which aims to deliver a robust script-independent solution optimized for both Kannada and English handwritten text. The system incorporates Local Sample-Weighted Multiple Kernel Clustering (LSMKC) for segmentation, Adaptive Multi-scale Gaussian Co-occurrence Filtering (AMGCF) for noise-free preprocessing, and Feature-Affine Residual Network (FA-ResNet) for enhanced feature extraction. A Unified Residual RNN optimized using the Secretary Bird Optimization Algorithm (SBOA) further refines character sequence prediction through the CTC loss function.

The major contributions of this study are:

- A TrOCR-URRNN-MLD framework for multilingual handwritten text recognition.
- Enhanced segmentation, preprocessing, and feature extraction through LSMKC, AMGCF, and FA-ResNet modules.
- Integration of URRNN+SBOA+CTC for robust and optimized recognition across multiple languages.

II. PROPOSED METHOD

Handwritten images from the English IAM OCR, Kannada Char74k, and multilingual datasets are processed through Local Sample-Weighted Multiple Kernel Clustering (LSMKC) for line, word, and character segmentation. The segmented images undergo Adaptive Multi-scale Gaussian Co-occurrence Filtering (AMGCF) for noise removal, contrast enhancement, and normalization. Feature-Affine Residual Network (FA-ResNet) extracts detailed spatial and geometric features, which are processed by the Unified Residual Recurrent Neural Network (URRNN) to learn sequential dependencies. The Secretary Bird Optimization Algorithm (SBOA) further refines the URRNN parameters, ensuring robust and accurate multilingual recognition. Figure 1 illustrates the overall workflow.

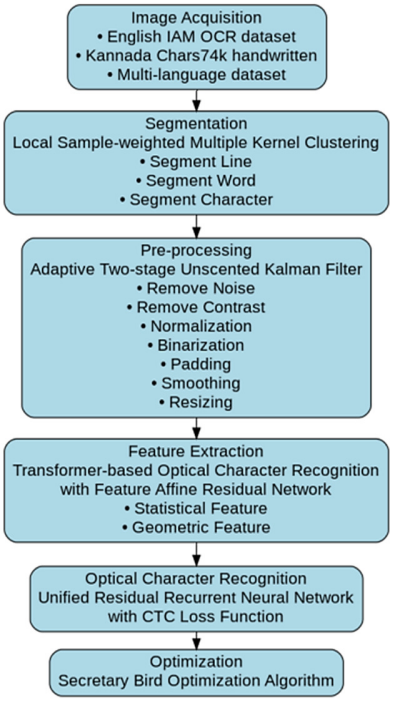

Fig. 1. Block diagram of the TrOCR-URRNN-MLD technique.

A. Data Acquisition

The input images are sourced from the English IAM OCR dataset [16] and the Kannada Char74k handwritten words dataset [17], combined into a Multilingual dataset. The English IAM OCR Dataset is a benchmark corpus for handwritten English text recognition, comprising ~100,000 images of lines, words, and characters from 657 writers. Its diverse cursive handwriting and detailed annotations make it ideal for handwriting recognition, segmentation, and writer

identification. The Kannada Char74k Dataset is derived from Chars74K, containing over 657 character classes (vowels, consonants, compound characters) with 25–50 samples per class, totaling ~100,000 images. Collected from multiple native writers and stored in PNG format, it captures diverse handwriting styles for Kannada OCR development.

Combining IAM OCR and Kannada Char74k, the Multilingual Dataset contains ~200,000 images in English and Kannada scripts from 657 writers. It supports multilingual OCR, script-based segmentation, and writer-independent recognition. The dataset is split into 45% training, 25% testing, and 30% validation.

The IAM and Char74k datasets are publicly available and used strictly for academic research. No personal identities or private information are included; therefore, ethical approval is not required.

B. Segmentation Using Local Sample-Weighted Multiple Kernel Clustering (LSMKC)

Segmentation of lines, words, and characters is performed using LSMKC [18], an advanced technique for text image segmentation. LSMKC uses multiple kernel functions and local weighting to capture diverse structural patterns and adapt to intensity variations, enabling robust segmentation with minimal manual intervention while effectively handling complex handwriting structures. The image enhancement process for clearer segmentation is formalized in

$$\chi_q = \frac{\partial_q}{\sqrt{\sum_{q=1}^{w} \partial_q^2}} \quad (1)$$

where χ_q denotes the clarity of the image and ∂_q indicates the variations in it. Neighborhood kernel construction enhances clustering by emphasizing local relationships among image pixels rather than global similarity. Kernel values are computed based on a sample's distance to its neighbors, preserving fine-grained variations in the image, as formalized in

$$H_a = max(\widehat{H}_{a,:} + \delta_a r_s^W) \quad (2)$$

where H_a denotes the neighborhood kernel construction, $\widehat{H}_a$ signifies the neighborhood kernel construction, and r_s^W is the multiple kernel size from an image. For line segmentation, morphological operators enhance line structures, and K-Means clustering aggregates connected components into lines, effectively handling skewed handwriting and non-uniform spacing (3).

$$v_{ab}^* = max(-\frac{f_{ab}}{2(\beta+\lambda_a)} + \xi) \quad (3)$$

where v_{ab}^* signifies the segment lines from an image, f_{ab} indicates the kernel value between images, β denotes the total area for dividing the images, λ_a represents multi-view clustering indications, and ξ represents the capturing of nonlinear structures from the kernel. In LSMKC, word and character segmentation separate connected components into individual units, improving recognition accuracy (4).

$$\vartheta = \frac{1}{p} + \frac{1}{2p(\delta+\eta_a)} \sum_{p=1}^{b=2} f_{ab} \quad (4)$$

where ϑ indicates the segmented word and character, p represents the localized kernel, δ denotes the normalized image, and η_a signifies the localized kernels preserved. By using LSMKC, the image is automatically divided into regions containing text, separating words and characters from the image background. The segmented images are given to the preprocessing phase. Figure 2 illustrates segmented images from the Multilingual Dataset.

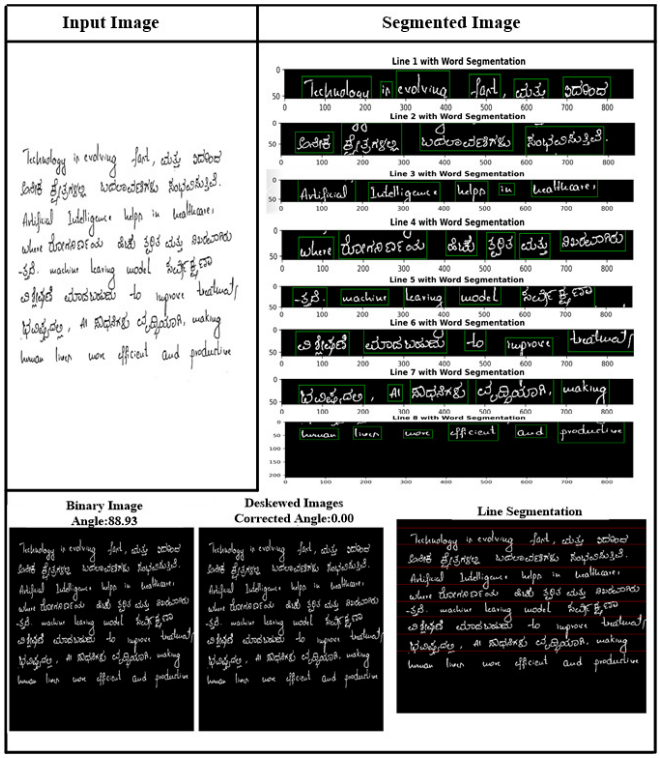

Fig. 2. Segmented images from the Multilingual dataset.

C. Preprocessing Using the Adaptive Two-Stage Unscented Kalman Filter (ATUKF)

Preprocessing employs the ATUKF [19] to remove noise, enhance contrast, normalize, binarize, pad, smooth, and resize segmented images. The ATUKF adaptively refines the filtering process, mitigating outliers and inconsistencies while preserving structural integrity. Each image is represented by an initial state vector corresponding to pixel values, capturing all necessary variables to define its condition before processing. This ensures improved clarity and quality of input images for subsequent feature extraction, as formalized in

$$v_h = k(a_h, n_h) + Ts_h + u_h h \geq \lambda \quad (5)$$

where v_h denotes the initial state value from the input images, k signifies the intensity values, a_h indicates the average value of the image, n_h represents the state vector and covariance matrix, T is the time taken for analysis, s_h indicates the image transformation function, and u_h designates the random variations in pixel values. Contrast enhancement in ATUKF is guided by intensity variations in the input image, refining pixel values to achieve a balanced distribution while preserving

visual quality. The filter dynamically adjusts its state estimation to reduce noise and restore ideal pixel values by leveraging spatial and intensity correlations. The refined pixel values are expressed by

$$\hat{G}_{l+1|l}^{pp} = \frac{1}{k-1}\sum_{j=l-k+2}^{l+1} \tilde{\varepsilon}_j \tilde{\varepsilon}_j^T \quad (6)$$

where $\hat{G}_{l+1|l}^{PP}$ indicates the pixel values from the image, Σ signifies the accumulated pixel-related error contributions, j represents the j^{th} dimension of the trends, k indicates the uncertain noise in the stroke, $\tilde{\varepsilon}_j$ identifies the observed and estimated pixel intensity values at index j, and T denotes the transpose of the error vector in the noisy image. To preserve consistency across images, initializing the value of an image entails setting pixel intensity values to a standard range. This process enhances visual quality, removes noise, contrasts and normalizes the effects of lighting variations, and ensures uniform brightness and contrast. Then, the pixel values of the image are normalized as

$$D_{l+1} = \left(\hat{G}_{l+1|l}^{PP} - Y\right)U_f^{-1} \quad (7)$$

where D_{l+1} indicates the normalized pixel value, Y represents the visual quality of the image, and U_f^{-1} indicates the gray level in an image. In ATUKF, the resizing process transforms input images to uniform dimensions while preserving aspect ratios through binarization, padding, and smoothing. The filter dynamically adjusts pixel values to maintain structural integrity. The processed images are reconstructed to ensure consistency for subsequent analysis, as formalized in

$$f_{h+1}^{a} = W_{h+1}^{-1}(\hat{G}_{l+1|l}^{PP} - W_{h+1}R^{a}W_{h+1}^{f} - U_a)(m_a W_{h+1}^{f})^{-1} \quad (8)$$

where f_{h+1}^{a} denotes the resized images, W_{h+1}^{-1} represents the scaling matrix, R^a is the rearranging image, m_a signifies the analytical value of the image, and U_a indicates the diagonal value of the image. Noise removal enhances image quality by eliminating unwanted distortions such as random variations in brightness and color. By minimizing these pixel values, noise is effectively removed from the entire input image:

$$z_{h+1|h}^{b} = {}^{h+1|h} z_{h|h}^{b} + f_{h+1}^{b} m_b \quad (9)$$

where $z_{h+1|h}^{y}$ represents the removal of noise, and b denotes the minimized pixel values. Finally, the images are resized, and the noise is removed by utilizing the ATUKF method. Then the preprocessed images are given to the feature extraction phase.

D. Feature Extraction Utilizing Transformer-Based OCR with Feature Affine Residual Network

Feature extraction leverages transformer-based OCR integrated with FA-ResNet [20] to capture both statistical and geometric features from image data. FA-ResNet enhances texture representation by adaptively modulating feature maps with affine transformations and robust residual connections, ensuring stable training and multi-scale feature learning. Input images are first resized and normalized to a standard resolution for consistency across datasets, and then processed through an initial convolutional layer to extract foundational patterns, which serve as building blocks for higher-level texture characteristics, as formalized in

$$\psi = \frac{1}{2E \times L \times O}\sum_{j=1}^{O}\sum_{k=1}^{L}\left(\{\hat{g}_{jk}\} - \nu_j\right)^2 \quad (10)$$

where ψ represents the variance of the sample, E denotes the dimensionality of the parameter, L signifies the selected neighbors, O represents the number of points, j and k represent the convolutional layers, $\hat{g}_{jk}$ represents the linear parameter of the classified image, ν represents the variables, and j represents the input value of the image. FA-ResNet reduces the dimensionality of features by integrating attention mechanisms within residual blocks. It selectively emphasizes important spatial and channel-wise features, enabling compact and discriminative representations in an image as

$$\{w_{ab}\} = \nu \oplus \frac{\{\hat{e}_{ab}] - \sigma_a}{\kappa + \rho} + \chi \quad (11)$$

where w_{ab} signifies the dimensionality of the feature, ν indicates the variation of the image, $\hat{e}_{ab}$ represents the normalized features, σ_a indicates the enhanced features, κ signifies max pooling, and ρ is the local and global texture pattern.

In FA-ResNet, an aggregation function is used to combine feature maps from various layers of the network to form a unified and informative image representation. This function includes operations such as concatenation and element-wise addition, enabling the model to integrate multi-scale and multi-level features to extract statistical and geometric features from the image. By aggregating these features, the network captures both local and global texture patterns as

$$H_j = B\left(\gamma\left(g_{jk}\right)\right) \quad (12)$$

where B represents the extracted features, γ represents the sequence of mapping operations, and g_{jk} represents the linear parameters of the image. Finally, the statistical and geometric features are extracted from the image using FA-ResNet. After completing feature extraction, the extracted features are given to OCR.

E. OCR Utilizing the Unified Residual Recurrent Neural Network (URRNN) with CTC Loss Function

The URRNN with CTC loss [21] models temporal dependencies in handwritten text sequences without requiring explicit character segmentation. It combines residual CNNs for capturing detailed stroke patterns in English and Kannada scripts with sequential learning, enabling effective recognition of variable-length and irregular handwriting. The CTC loss aligns predicted sequences with ground truth, providing flexibility for diverse handwriting styles. Input grayscale images are resized to a constant height while preserving aspect ratio, ensuring uniform input for downstream processing. This integrated architecture supports accurate multilingual OCR by maintaining spatial consistency and robust feature representation as

$$r_s = \{f_{qxe,s}, f_{q\varpi,s}\} \quad (13)$$

where r_s denotes the input layer, $f_{qxe,s}$ indicates the residual connection, and $f_{q\varpi,s}$ represents the time distributed layer. The convolutional layers extract hierarchical spatial features from handwritten images, while residual connections preserve information across layers and mitigate gradient vanishing, enabling deeper network training. Each residual block comprises convolution, batch normalization, and ReLU activation, with skip connections to ensure stable and efficient feature learning across diverse handwriting styles, as

$$t_s = \vartheta(\varpi^{(t)} R_s + V^{(t)} g_{s-1} + g^{(t)}) \quad (14)$$

where t_s denotes the convolutional layer, ϑ signifies the hierarchical spatial features, $\varpi^{(t)}$ indicates the trainable weight matrix, R_s represents the ReLU activation function, g designates the global average pooling operation, and $V^{(t)}$ is the gated recurrent unit layer. The recurrent layers, often implemented as bidirectional, capture the sequential nature of handwritten text by processing extracted spatial features as temporal sequences. They model contextual dependencies from both past and future positions, improving recognition of complex cursive handwriting and compound characters as

$$g_s = (1 - t_s) g_{s-1} + t_s * g_s' \quad (15)$$

where t_s denotes the recurrent layers, $*$ indicates the element-wise multiplication, and g_s is the bidirectional temporal capture. The fully connected layer acts as a linear classifier, converting feature vectors at each time step into a probability distribution over English and Kannada characters. It enables the model to predict sequences consistent with the temporal structure of handwritten text as

$$loss = -\sum_{\bar{S}}^{a=1} q_a \, log(\hat{q}_a) \quad (16)$$

where q_a indicates the fully connected layer and $\hat{q}_a$ denotes the total number of character classes. The CTC loss layer enables training without requiring pre-segmented character alignment, handling variable-length output sequences. It computes the probability of correct label sequences by summing over all valid alignments between predictions and ground truth, making it particularly effective for OCR of English and Kannada scripts as

$$\beta_{new} == \beta_{old} + \varsigma(\frac{gloss}{g\beta}) \quad (17)$$

where β denotes the prediction of optical character recognition, $gloss$ indicates the CTC loss layer, and ς is the character boundaries. The URRNN with CTC loss function is employed for the OCR stage. The URNN was selected due to its pertinence, convenience, and AI-dependent optimization approach.

F. Optimization Using the Secretary Bird Optimization Algorithm (SBOA)

SBOA [22] is a novel meta-heuristic technique employed to optimize model parameters, enhancing accuracy and reducing overfitting. SBOA balances exploration and exploitation to efficiently search the parameter space, preventing the model from getting trapped in local minima. It improves convergence speed, maintains solution diversity, and adapts robustly to high-dimensional optimization problems, outperforming conventional methods prone to premature stagnation. In the proposed OCR framework, SBOA fine-tunes URRNN parameters, optimizing weights to achieve higher accuracy while reducing computational time. The algorithm follows a series of principal stages, which guide its iterative search for globally optimal solutions, ensuring robust and reliable recognition performance for multilingual handwritten text.

1) Step 1: Initialization

SBOA is a population-based metaheuristic approach, in which each secretary bird is a member of the algorithm's population. Here, the positions of every secretary bird in the search space determine the values of the decision variables used to compute the MASNN weight parameters.

$$h = \begin{bmatrix} h_{11} & h_{12} & h_{13} & \cdots & h_{1h} \\ h_{21} & P_{22} & h_{23} & \cdots & h_{2h} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{nd1} & h_{d2} & h_{d3} & \cdots & h_{dh} \end{bmatrix} \quad (18)$$

where h represents the secretary bird group and h_{dh} represents the group members.

2) Step 2: Random Generation

Input parameters are made at random. Dependent on particular hyperparameter conditions, r_s and g_s parameters are created at random through the SBOA method.

3) Step 3: Fitness Function

A random solution is generated using initial evaluations. Parameter optimization values are evaluated for optimizing the weight parameters of the URRNN as

$$FitnessFunction = optimizing[r_s \; and \; g_s] \quad (19)$$

4) Step 4: Hunting Strategy of Secretary Birds

In SBOA, a secretary bird is a predator that stalks by rapidly scanning open plains and quietly sneaking up on its prey, mainly snakes and small mammals. It makes rapid, high-accuracy stomps with its strong legs to paralyze or kill the target from a safe distance. The ground-level hunting process involves speed and adaptive movement as

$$w_a = \begin{cases} w_a^{new}, & \text{if } r_s Z_a^{new,q1} < Z_a \\ w_a, & \text{else} \end{cases} \quad (20)$$

where w_a represents a new state of the secretary bird, $w_a^{new,q1}$ signifies random candidate solutions within the initial stage iteration, w_a^{new} denotes the new state of a secretary bird in the initial stage, and Z_a signifies a randomly generated array.

$$levy(P) = k * \frac{u * \varepsilon}{|v|^{\frac{1}{\pi}}} \quad (21)$$

where $levy(P)$ represent the fight distribution function, k signifies a fixed constant value, $u * \varepsilon$ denotes the standard normal distribution, and $|v|^{\frac{1}{\pi}}$ signifies the update of the secretary bird's position.

5) Step 5: Escape Strategy of Secretary Bird for Optimization

Dynamic evasion strategies are used against adversaries. This method guarantees resistance to attempts to trick the prediction system, strengthening the model through adaptive approaches, effectively preserving prediction robustness.

$$w_a = \begin{cases} w_a^{new}, \text{if } Z_a^{new,q2} g_s < Z_a \\ w_a, \quad \text{else} \end{cases} \tag{22}$$

where $Z_a^{new,q2}$ denotes random candidate solutions in the second stage iteration, w_a^{new} denotes a new state of a secretary bird in the initial stage, and g_s signifies bidirectional temporal capture. The random selection is given by

$$H = round\,(1 + qand(1,1)) \tag{23}$$

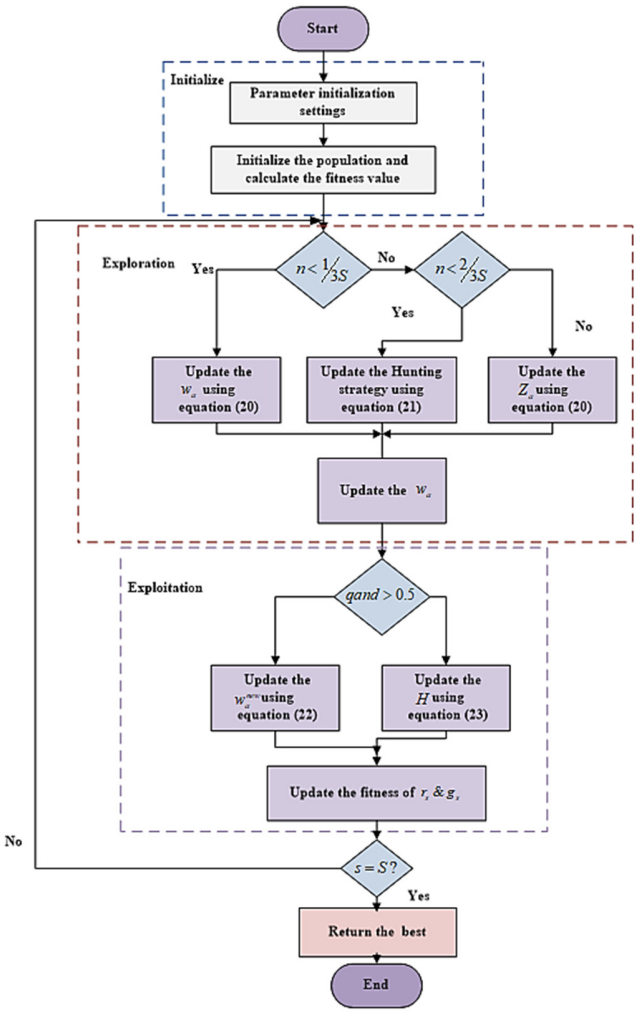

Fig. 3. Flowchart of SBOA for optimizing URRNN parameters.

6) Step 6: Termination

The weight parameter values r_s g_s of the URRNN are optimized using SBOA, repeating from step 3 until it obtains its halting criteria. Then, TrOCR-URRNN-MLD effectively assesses the accuracy of recognizing the optical character, reducing Word Error Rate (WER).

III. RESULTS AND DISCUSSION

The proposed TrOCR-URRNN-MLD model was implemented in Python using Keras and TensorFlow on a Windows 10 laptop with NVIDIA Tesla P100 GPU (16 GB RAM), achieving a runtime of 760.3 s. The model was evaluated on the English IAM OCR, Kannada Char74k, and the combined multilingual dataset. Performance was compared with existing methods, including DL-SI-CRN, SVM-MCR, and HCR-DSRNN-MaxEnt, using standard OCR metrics such as Accuracy, Precision, Recall, F1-score, WER, Character Error Rate (CER), and Text Similarity (TS). The model was trained using a batch size of 32 for 150 epochs. The learning rate was set to 0.0001, and the training employed the Adam optimizer. A dropout rate of 0.3 was applied to prevent overfitting. The input image size used for the model was 64×256 pixels. Table I shows a comprehensive analysis of performance metrics of various OCR techniques, highlighting the superior performance of the TrOCR-URRNN-MLD method across all evaluation metrics.

TABLE I. PERFORMANCE ANALYSIS OF VARIOUS OCR TECHNIQUES

Ref.	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	WER (%)	CER (%)	TS
[7]	93.45	94.90	92.45	90.45	31.67	30.90	89.43
[10]	91.84	90.70	94.90	94.70	43.61	25.91	91.07
[11]	94.60	93.76	93.37	92.64	22.43	43.93	82.39
[12]	91.66	85.50	89.42	88.25	28.81	38.92	95.10
[14]	94.98	90.74	90.31	93.28	37.43	41.93	85.87
Proposed	97.68	97.83	97.85	97.73	16.60	8.04	97.7

The proposed model achieved 97.68% accuracy, 97.83% precision, 97.85% recall, and 97.73% F1-score, with a WER of 16.60% and a CER of 8.04%, outperforming existing approaches that report accuracies below 95% and higher error rates. A text similarity score of 97.7% confirms high semantic fidelity and reliable recognition across English and Kannada scripts. Figure 4 illustrates predicted images using URRNN. The proposed model effectively captures complex handwriting patterns while maintaining computational efficiency. Although deep model training on high-resolution images is computationally intensive, the proposed approach demonstrates lower processing time than comparable approaches, making it suitable for real-time multilingual OCR applications.

IV. CONCLUSION

The proposed TrOCR-URRNN-MLD model effectively performs multilingual handwritten OCR using the English IAM OCR, Kannada Char74k, and combined Multilingual datasets. By integrating FA-ResNet for feature extraction, URRNN for sequential modeling, and SBOA for parameter optimization, the system achieves accurate recognition across English and Kannada scripts while reducing computation time and accelerating convergence. Compared to existing methods, such as DL-SI-CRN, SVM-MC, and HCR-DSRNN-MaxEnt, TrOCR-URRNN-MLD attains 6–11% lower WER and CER

and 6–8% higher text similarity, demonstrating superior performance.

Fig. 4. Predicted Images using URRNN

Future work will explore hybrid optimization techniques and real-time OCR deployment using edge computing and IoT-enabled devices to further enhance accessibility and efficiency. Although the proposed model achieves strong performance, training requires high computational resources and is currently tested only on Kannada and English datasets. Future research will explore extensions to more language families and lightweight deployment.

REFERENCES

[1] D. Coquenet, C. Chatelain, and T. Paquet, "End-to-End Handwritten Paragraph Text Recognition Using a Vertical Attention Network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no.

1, pp. 508–524, Jan. 2023, https://doi.org/10.1109/TPAMI.2022.3144899.

[2] K. M. O. Nahar *et al.*, "Recognition of Arabic Air-Written Letters: Machine Learning, Convolutional Neural Networks, and Optical Character Recognition (OCR) Techniques," *Sensors*, vol. 23, no. 23, Jan. 2023, Art. no. 9475, https://doi.org/10.3390/s23239475.

[3] B. R. Kavitha and C. Srimathi, "Benchmarking on offline Handwritten Tamil Character Recognition using convolutional neural networks," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 4, pp. 1183–1190, Apr. 2022, https://doi.org/10.1016/j.jksuci.2019.06.004.

[4] S. Vijayalakshmi, K. R. Kavitha, B. Saravanan, R. Ajaybaskar, and M. Makesh, "Handwritten Character Recognition for Tamil Language Using Convolutional Recurrent Neural Network," in *Inventive Systems and Control*, 2022, pp. 369–384, https://doi.org/10.1007/978-981-19-1012-8_25.

[5] M. Dhiaf, A. C. Rouhou, Y. Kessentini, and S. B. Salem, "MSdocTr-Lite: A lite transformer for full page multi-script handwriting recognition," *Pattern Recognition Letters*, vol. 169, pp. 28–34, May 2023, https://doi.org/10.1016/j.patrec.2023.03.020.

[6] J. Mukherjee and U. Roy, "A Low Resource Multi-lingual Simultaneous Script Identification and Text Recognition Model," *SN Computer Science*, vol. 5, no. 6, July 2024, Art. no. 740, https://doi.org/10.1007/s42979-024-03107-6.

[7] V. K. Chauhan, S. Singh, and A. Sharma, "HCR-Net: a deep learning based script independent handwritten character recognition network," *Multimedia Tools and Applications*, vol. 83, no. 32, pp. 78433–78467, Sept. 2024, https://doi.org/10.1007/s11042-024-18655-5.

[8] B. Rabhi, A. Elbaati, H. Boubaker, U. Pal, and A. M. Alimi, "Multi-lingual handwriting recovery framework based on convolutional denoising autoencoder with attention model," *Multimedia Tools and Applications*, vol. 83, no. 8, pp. 22295–22326, Mar. 2024, https://doi.org/10.1007/s11042-023-16499-z.

[9] T. Hasan, Md. A. Rahim, J. Shin, S. Nishimura, and Md. N. Hossain, "Dynamics of Digital Pen-Tablet: Handwriting Analysis for Person Identification Using Machine and Deep Learning Techniques," *IEEE Access*, vol. 12, pp. 8154–8177, 2024, https://doi.org/10.1109/ACCESS.2024.3352070.

[10] S. P. Ramteke, A. A. Gurjar, and D. S. Deshmukh, "A Novel Weighted SVM Classifier Based on SCA for Handwritten Marathi Character Recognition," *IETE Journal of Research*, vol. 68, no. 2, pp. 845–857, Mar. 2022, https://doi.org/10.1080/03772063.2019.1623093.

[11] N. Tripathi and P. S. Patheja, "Offline handwritten character recognition with nomograph-based IMVO feature mining with DSRNN-MaxEnt classification," *Sādhanā*, vol. 48, no. 4, Nov. 2023, Art. no. 272, https://doi.org/10.1007/s12046-023-02327-5.

[12] R. Malhotra and M. T. Addis, "End-to-End Historical Handwritten Ethiopic Text Recognition Using Deep Learning," *IEEE Access*, vol. 11, pp. 99535–99545, 2023, https://doi.org/10.1109/ACCESS.2023.3314334.

[13] B. A. Tuama and F. Mohamed, "A Systematic Literature Review of Deep Learning Methods for Handwritten Text Recognition in Historical Arabic Manuscripts," *Engineering, Technology & Applied Science Research*, vol. 15, no. 4, pp. 25772–25782, Aug. 2025, https://doi.org/10.48084/etasr.12123.

[14] A. Moudgil, S. Singh, V. Gautam, S. Rani, and S. H. Shah, "Handwritten devanagari manuscript characters recognition using capsnet," *International Journal of Cognitive Computing in Engineering*, vol. 4, pp. 47–54, June 2023, https://doi.org/10.1016/j.ijcce.2023.02.001.

[15] B. Kada, A. Mohammed, and B. Abdelmajid, "An Optimized Approach for Handwritten Arabic Character Recognition based on the SVM Classifier," *Engineering, Technology & Applied Science Research*, vol. 15, no. 2, pp. 22232–22238, Apr. 2025, https://doi.org/10.48084/etasr.9292.

[16] "English - IAM OCR dataset." Kaggle, [Online]. Available: https://www.kaggle.com/datasets/python16/english-iam-ocr-dataset.

[17] "kannada Char74k handwritten words dataset." Kaggle, [Online]. Available: https://www.kaggle.com/datasets/mcpython/kannada-char74k-handwritten-words-dataset.

[18] L. Li *et al.*, "Local Sample-Weighted Multiple Kernel Clustering With Consensus Discriminative Graph," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 2, pp. 1721–1734, Oct. 2024, https://doi.org/10.1109/TNNLS.2022.3184970.

[19] X. Gong, Z. Hou, A. Ma, Y. Zhong, M. Zhang, and K. Lv, "An Adaptive Multiscale Gaussian Co-Occurrence Filtering Decomposition Method for Multispectral and SAR Image Fusion," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 8215–8229, 2023, https://doi.org/10.1109/JSTARS.2023.3296505.

[20] L. Zhan, W. Li, and W. Min, "FA-ResNet: Feature affine residual network for large-scale point cloud segmentation," *International Journal of Applied Earth Observation and Geoinformation*, vol. 118, Apr. 2023, Art. no. 103259, https://doi.org/10.1016/j.jag.2023.103259.

[21] A. Al-Malahi, A. Farhan, H. Feng, O. Almaqtari, and B. Tang, "An intelligent radar signal classification and deinterleaving method with unified residual recurrent neural network," *IET Radar, Sonar & Navigation*, vol. 17, no. 8, pp. 1259–1276, 2023, https://doi.org/10.1049/rsn2.12417.

[22] Y. Fu, D. Liu, J. Chen, and L. He, "Secretary bird optimization algorithm: a new metaheuristic for solving global optimization problems," *Artificial Intelligence Review*, vol. 57, no. 5, Apr. 2024, Art. no. 123, https://doi.org/10.1007/s10462-024-10729-y.