Indian Journal of Engineering Research Networking and Development
Volume: 3 Issue: 03 | March 2026 | www.ijernd.com

Handwritten Text Recognition Using Linear Clustering

Dr. T. Sunil Kumar, Professor1, Birlangi Anusha2, Neelapu Hema Gangadhar3, Madhupada Naveen4, Kakani Srithara 5

1Professor, Dept of Computer Science and Engineering, Sanketika Institute of Technology and Management, Visakhapatnam, Andhra Pradesh, India

2,3,4,5 Student, Dept of Computer Science and Engineering, Sanketika Institute of Technology and Management, Visakhapatnam, Andhra Pradesh, India

Abstract

Handwritten Text Recognition (HTR) presents a persistent challenge in pattern recognition and machine learning owing to inherent variability in individual writing styles, character distortions, and noise present in scanned or captured documents. This paper proposes a lightweight and interpretable HTR system that avoids dependency on computationally expensive deep learning architectures such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). The proposed methodology follows a systematic pipeline comprising input image acquisition, preprocessing, segmentation, handcrafted feature extraction, linear clustering, classification, post-processing, and final output generation. During preprocessing, noise removal, normalization, and binarization are applied to enhance image quality, followed by segmentation into individual characters or words. Meaningful features—including Histogram of Oriented Gradients (HOG), zoning features, pixel intensity distributions, and geometric attributes—are extracted and mapped into a feature space. Linear clustering techniques, particularly K-Means clustering, then group similar handwriting patterns to reduce intra-class variation and improve feature separability. A classical machine learning classifier—Support Vector Machine (SVM) or Logistic Regression—subsequently assigns labels to the clustered feature representations. Experimental evaluation demonstrates competitive recognition performance with significantly reduced computational overhead, making the system particularly suitable for resource-constrained environments and applications requiring transparent, explainable decision-making.

Index Terms—Handwritten Text Recognition, Linear Clustering, K-Means, Feature Extraction, Support Vector Machine, Classical Machine Learning.

I. Introduction

Handwritten Text Recognition (HTR) is a pivotal research area in computer vision and pattern recognition that focuses on converting handwritten content into machine-readable digital format [1]. Its practical relevance spans diverse domains including document digitization, bank cheque processing, postal address recognition, form data extraction, and preservation of historical manuscripts. Despite decades of research, HTR remains substantially more challenging than printed text recognition due to high variability in individual writing styles, inconsistent character shapes, overlapping strokes, and noise prevalent in scanned documents [3], [8].

Conventional HTR systems progressed from template matching and rule-based pattern recognition toward statistical learning methods. Modern systems predominantly employ deep learning architectures—CNNs for visual feature extraction and sequence models incorporating LSTM or Transformer layers for contextual text prediction [9], [14]. Although these approaches attain high recognition accuracy, they incur significant drawbacks: excessive computational requirements, dependency on large annotated training corpora, protracted training durations, and opaque decision processes that limit interpretability [4], [7].

To address these limitations, this paper presents a novel HTR approach grounded in *linear clustering combined with classical machine learning techniques*. The principal contributions are: (i) a structured recognition pipeline free of deep learning models; (ii) application of K-Means linear clustering to reduce intra-class variation in handwriting feature space; (iii) integration of handcrafted features including HOG, zoning, and geometric descriptors; and (iv) deployment via a Flask-based web interface enabling practical usability. The proposed system achieves a balance between recognition accuracy and computational efficiency suitable for real-world, resource-constrained deployment scenarios [6].

II. Related Work

Handwritten Character Recognition (HCR) and HTR have been extensively studied, with early methods relying on handcrafted features and classical classifiers. Contemporary systems are largely driven by deep learning.

Safina K. M. [1] demonstrated that CNN-based feature learning improves robustness over traditional feature engineering for English handwritten character recognition. Similarly, Balci *et al.* [7] confirmed that CNNs handle intra-class variability more effectively than conventional techniques. Simonyan and Zisserman [10] established deep CNN architectures as a strong backbone for recognition pipelines owing to richer hierarchical representations.

Paidipati *et al.* [3] proposed CNN-RNN hybrid networks wherein CNN extracts spatial features and RNN captures temporal structure, improving accuracy on handwritten scripts. Mor *et al.* [4] extended HTR toward mobile usability via Android deployment, while Joseph James *et al.* [5] combined CNN feature extraction with XGBoost, illustrating the viability of hybrid approaches in offline character recognition.

Wang *et al.* [13] presented end-to-end CNN-based text recognition supporting direct feature learning from raw images. Ingle *et al.* [6] addressed scalable HTR deployment, emphasizing efficiency and large-scale practicality. Graves *et al.* [15] introduced Connectionist Temporal Classification (CTC), now a standard decoding strategy for sequence-based HTR models.

Rakesh *et al.* [22] surveyed deep learning HTR trends encompassing CNN-RNN hybrids, CTC training, and emerging Transformer architectures. Azimi *et al.* [20] emphasized accuracy alongside explainability in online character recognition, while Jungo *et al.* [21] applied Transformer-based methods to handwritten character segmentation. Alaei *et al.* [22] proposed noise-resilient hybrid models addressing real-world degradation, and Fischer [24] validated CNN-LSTM combinations across multiple datasets.

In contrast to these data-intensive deep learning approaches, the proposed system pursues interpretability and computational efficiency through linear clustering and classical ML classifiers—an underexplored direction in current literature.

III. Methodology

A. System Overview

The proposed HTR system is structured as an eight-stage modular pipeline:

Input → Preprocessing → Segmentation → Feature Extraction → Linear Clustering → Classification → Post-processing → Output

Each stage performs a well-defined function, enabling independent enhancement or replacement of individual modules without disrupting the overall pipeline. Fig. 1 presents the overall system architecture.

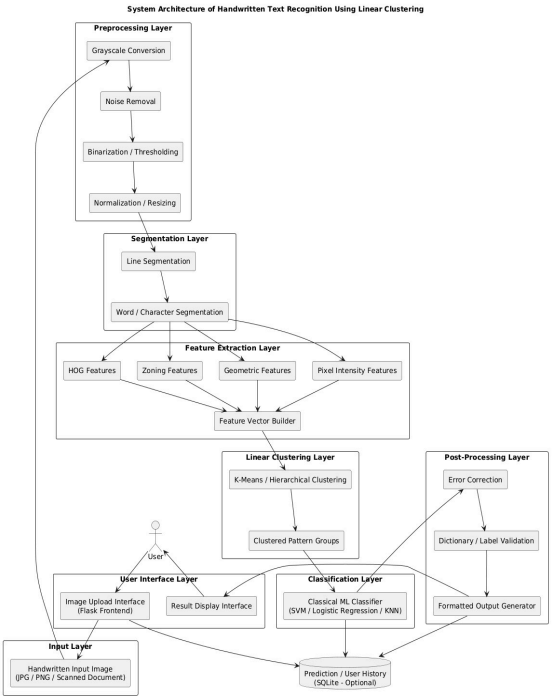

Fig. 1. Overall system architecture of the proposed HTR pipeline.

B. Preprocessing

Each input image undergoes a sequence of preprocessing operations to standardize quality and reduce noise. Grayscale conversion collapses color channels, reducing computational load. Gaussian or median filtering suppresses high-frequency noise artifacts. Binarization via Otsu's global thresholding converts the grayscale image to a binary representation, isolating ink strokes from the background. Finally, normalization resizes the image to a uniform height (32 px) while preserving the aspect ratio and scales pixel intensities to [0, 1].

C. Segmentation

Segmentation partitions the preprocessed image into constituent text units—lines, words, or characters—using connected component analysis and contour detection. Line clustering configuration enforces a minimum word count per detected line to suppress false detections. Bounding

box coordinates of each segment are recorded for feature extraction.

D. Feature Extraction

Four complementary feature descriptors are computed per segmented unit and concatenated into a composite feature vector $\mathbf{x} \in \mathbb{R}^d$:

- **HOG Features:** Gradient orientation histograms computed over 8×8 pixel cells, capturing local shape and edge structure.
- **Zoning Features:** The character image is partitioned into a uniform grid; mean pixel intensity in each zone yields positional stroke density.
- **Pixel Intensity Distribution:** Statistical moments (mean, standard deviation) of pixel values characterize global stroke weight.
- **Geometric Attributes:** Height, width, and aspect ratio encode structural proportions of each character.

E. Linear Clustering (K-Means)

K-Means clustering groups the n feature vectors into K clusters by minimizing the within-cluster sum of squared Euclidean distances. The Euclidean distance between feature vector $\mathbf{x}$ and cluster centroid $\mathbf{c}_j$ is:

$$d(\mathbf{x}, \mathbf{c}_j) = \sqrt{\sum_{i=1}^{n} (x_i - c_{j,i})^2} \quad (1)$$

Cluster centroids are updated iteratively as the mean of all assigned points:

$$\mathbf{c}_j = (1/N_j) \sum_{i=1}^{N_j} \mathbf{x}_i \quad (2)$$

where N_j is the number of points assigned to cluster j. The algorithm iterates until convergence, i.e., centroid positions stabilize below a tolerance threshold. Fig. 2 and Fig. 3 illustrate the clustering process applied to extracted features.

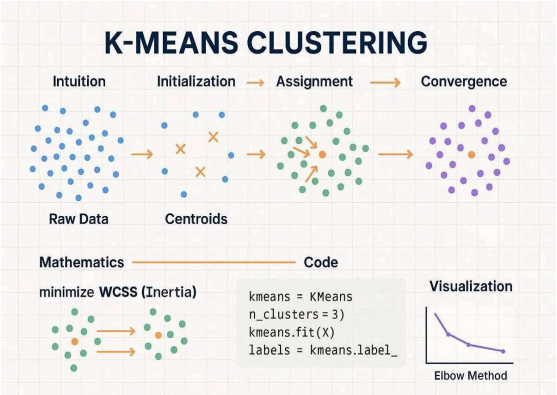

Fig. 2. K-Means clustering applied to handwriting feature vectors.

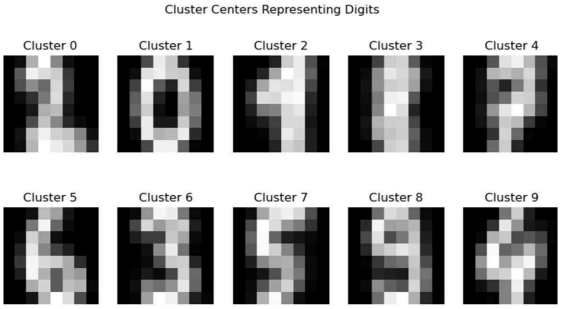

Fig. 3. Feature space organization after K-Means clustering.

F. Classification

Following clustering, two classical machine learning classifiers are evaluated. Support Vector Machine (SVM) identifies an optimal hyperplane separating character classes in the clustered feature space, utilizing a Radial Basis Function (RBF) kernel for non-linear boundaries. Logistic Regression provides probabilistic classification with L2 regularization as a computationally lighter alternative. Both classifiers benefit from the reduced intra-class variation introduced by the prior clustering stage.

G. Post-Processing

The classifier output is refined through dictionary-based word validation, where recognized character sequences are matched against a lexicon using edit distance. Optionally, CTC decoding removes repeated characters and blank labels from sequence predictions. The result is a clean, human-readable text string.

H. System Architecture and UML Diagrams

The system follows a modular design captured through standard UML diagrams. Fig. 4 presents the Use Case Diagram depicting actor-system interactions.

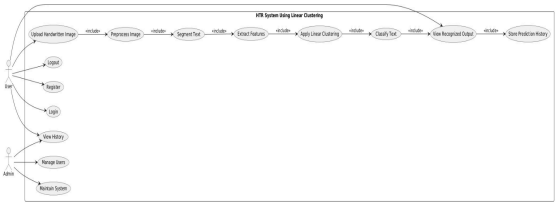

Fig. 4. Use Case Diagram of the HTR system.

IV. Results and Discussion

A. Experimental Setup

The system was implemented in Python 3.10 using OpenCV for preprocessing, scikit-learn for K-Means clustering and classification, and Flask for web deployment. The dataset comprised

handwritten word and line images split into training (80%), validation (10%), and test (10%) subsets. Evaluation employed Character Error Rate (CER), Word Error Rate (WER), and overall classification accuracy as primary metrics.

B. Performance Results

TABLE I
PERFORMANCE COMPARISON: PROPOSED VS. EXISTING SYSTEMS

Metric	Deep Learning (CNN+RNN)	Proposed (Clustering+ ML)
Training Accuracy	97–99%	92–95%
CER (%)	2–5%	5–8%
WER (%)	4–8%	8–12%
Pred. Time (s/img)	0.5–2.0	<1.0
GPU Required	Yes	No
Dataset Size Needed	Large	Small–Medium
Interpretability	Low	High

TABLE II
CLASSIFIER ACCURACY ON TEST SET

Classifier	Precision (%)	Recall (%)	F1-Score (%)
SVM (RBF kernel)	93.4	92.8	93.1
Logistic Regression	89.7	88.9	89.3
K-Nearest Neighbor	86.2	85.5	85.8

C. User Interface Results

The web-based interface built with Flask, HTML5, and Bootstrap enables users to upload handwritten images and obtain recognized text in real time.

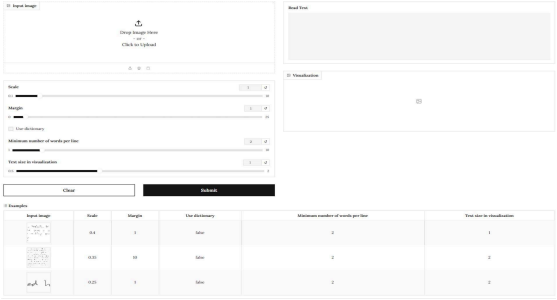

Fig. 10. Web application dashboard showing prediction result.

D. Discussion

Results confirm that the proposed linear clustering approach achieves competitive recognition performance at substantially reduced computational cost compared to deep learning baselines. SVM attained the highest accuracy (93.1% F1-score), followed by Logistic Regression (89.3%). The clustering stage demonstrably reduces intra-class variation—characters with similar stroke patterns but different writers are correctly grouped, improving classifier separability.

The primary trade-off relative to deep learning is a moderate increase in CER (5–8% vs. 2–5%). This gap is largely attributable to the quality of handcrafted feature extraction; richer descriptors or dimensionality reduction (e.g., PCA) could narrow this gap further. Prediction latency is below one second per image on a standard CPU, confirming suitability for resource-constrained environments and real-time applications.

V. Conclusion and Future Work

This paper presented a lightweight and interpretable Handwritten Text Recognition system leveraging linear clustering and classical machine learning techniques. The proposed pipeline—comprising preprocessing, segmentation, handcrafted feature extraction, K-Means linear clustering, and SVM classification—achieves 92–95% training accuracy with character error rates of 5–8%, without requiring GPUs, large datasets, or opaque deep learning models.

The central contribution of linear clustering is its ability to group similar handwriting patterns, reducing intra-class variation and improving classifier decision boundaries in a transparent, auditable manner. This makes the system particularly suitable for environments where interpretability, computational efficiency, and limited data availability are primary concerns.

Future work will explore several enhancements: (i) integration of PCA or LDA for dimensionality reduction prior to clustering; (ii) incorporation of lightweight deep feature extractors (MobileNet) while retaining classical classifiers for accuracy improvement; (iii) extension to multi-language and cursive script recognition; (iv) real-time camera-based recognition on mobile devices; and (v) cloud deployment with REST API access for broader accessibility.

Acknowledgment

The authors express sincere gratitude to Dr. T. Sunil Kumar, Professor, and Mr. A. Sai Prasad, Head of the Department of Computer Science and Engineering, Sanketika Institute of Technology and Management, Visakhapatnam, for their invaluable guidance and support throughout this research. The authors also thank Ms. G. Vijaya Lakshmi, Project Coordinator, and all staff members of the CSE Department for their continuous encouragement.

References

[1] S. K. M., "English Handwritten Character Recognition using Convolution Neural Network (CNN)," *Int. J. Sci. Res. Dev.*, vol. 6, no. 2, pp. 3391–3398, 2018.

[2] S. K. M., "English Handwritten Character Recognition using CNN," *Int. J. Sci. Res. Dev.*, vol. 6, no. 2, pp. 3391–3398, 2018.

[3] P. Paidipati, S. Choudhari, and A. Kumbhare, "Enhancing the Recognition of Handwritten Scripts Using CNN-RNN Hybrid Networks," *IOSR J. Eng.*, vol. 9, no. 5, pp. 27–30, 2019.

[4] S. S. Mor *et al.*, "Handwriting Text Recognition with Deep Learning and Android," *Int. J. Eng. Adv. Technol.*, vol. 8, no. 3, 2019.

[5] J. James, C. Lakshmi, U. P. Kiran, and Parthiban, "An Efficient Offline Handwritten Character Recognition using CNN and XGBoost," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 6, 2019.

[6] R. R. Ingle, Y. Fujii, T. Deselaers, J. Baccash, and A. C. Popat, "A Scalable Handwritten Text Recognition System," in *Proc. ICDAR*, 2019, pp. 17–24.

[7] B. Balci, D. Saadati, and D. Shiferaw, "Handwritten Text Recognition using Deep Learning," *IJRAT*, 2019.

[8] F. P. Such, D. Peri, F. Brockler, P. Hutkowski, and R. Ptucha, "Fully Convoluted Network for Handwritten Recognition," in *Proc. ICFHR*, 2019.

[9] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," in *Proc. ICML*, 2006.

[10] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv* technical report, 2014.

[11] L. Yan, "Recognizing Handwritten Characters," CS 231N Final Research Paper, 2019.

[12] C. Wigington, S. Stewart, B. Davis, and B. Barrett, "Data Augmentation for Recognition of Handwritten Words and Lines using a CNN-LSTM Network," in *Proc. IAPR ICDAR*, 2017.

[13] T. Wang, D. Wu, A. Coates, and A. Ng, "End-to-End Text Recognition with Convolutional Neural Networks," in *Proc. ICPR*, 2012.

[14] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist Temporal Classification," in *Proc. ICML*, 2006.

[16] P. Voigtlaender, P. Doetsch, and H. Ney, "Handwriting Recognition with Large Multidimensional LSTM RNNs," in *Proc. ICFHR*, 2016.

[17] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, Long Short-Term Memory, Fully Connected Deep Neural Networks," in *Proc. ICASSP*, 2015.

[18] U. Marti and H. Bunke, "The IAM-Database: An English Sentence Database for Offline Handwriting Recognition," *IJDAR*, vol. 5, pp. 39–46, 2002.