iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.82246

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

AI-Driven Hindi Handwritten Character Recognition Using Deep Convolutional Neural Network and Transformer Architectures

Sarang Ghatkar, Rekha Sharma
M.E Scholar, Associate Professor, Department of Computer Engineering Thakur College of Engineering & Technology, Kandivali (E), Mumbai – 400101, Maharashtra

Abstract— Hindi, one of the most widely spoken languages globally, poses distinctive challenges for automated character recognition due to its structurally complex Devanagari script, conjunct consonants (matras), and inherent diacritical marks. This paper proposes a novel hybrid deep learning framework that unifies Deep Convolutional Neural Networks (DCNN) with Transformer architectures for accurate Hindi character recognition. The model exploits the spatial feature extraction power of convolutional layers alongside the long-range dependency modeling capacity of Transformer self-attention. Evaluation on the IIIT-HW-Dev and Devanagari Character Dataset (DCD) demonstrates a state-of-the-art accuracy of 98.37%, substantially exceeding existing methods such as standalone CNNs (92.14%), ResNet-50 (94.63%), and Vision Transformer (95.81%). Detailed ablation studies validate the contribution of each architectural component, and robustness evaluations confirm the model's resilience across both handwritten and printed Hindi text.

Keywords— Hindi character recognition; Devanagari script; deep convolutional neural network; Vision Transformer; self-attention; optical character recognition; deep learning; hybrid architecture

I. INTRODUCTION

Automated recognition of handwritten and printed text represents a long-standing challenge in computer vision and pattern recognition. Considerable advances have been achieved for Latin-script languages; however, recognition systems for Indic scripts, particularly Hindi written in the Devanagari script, remain comparatively underdeveloped despite the enormous speaker population. Hindi ranks as the fourth most spoken language globally, with more than 600 million speakers, making automated Hindi character recognition (HCR) critically important for applications spanning historical document digitisation, real-time language translation, and accessibility tools for the visually impaired.

The Devanagari script is characterised by pronounced structural complexity. Unlike Latin characters, Devanagari employs consonant clusters (samyuktakshar), vowel diacritics (matras), and a distinctive horizontal headline (shirorekha) connecting letters within a word. These attributes render Devanagari considerably more challenging for conventional OCR systems, as characters frequently share overlapping visual features and exhibit substantial intra-class variation across diverse writing styles and typefaces.

Traditional approaches to Hindi character recognition relied on handcrafted feature descriptors such as Histogram of Oriented Gradients (HOG), Zernike moments, and structural features. While such methods yielded acceptable performance on constrained datasets, they generalised poorly to real-world handwritten text variability. The emergence of deep learning, particularly Convolutional Neural Networks (CNNs), represented a paradigm shift, enabling automatic feature learning directly from raw pixel data.

Transformer architectures, originally designed for natural language processing, have since been successfully adapted to computer vision tasks through Vision Transformers (ViT). The self-attention mechanism allows these models to capture long-range spatial dependencies between different image regions — a property particularly beneficial for recognising characters with complex spatial relationships. Pure Transformer models, however, demand large training datasets and substantial computational resources, motivating a hybrid design.

This work proposes a hybrid architecture that harnesses the complementary strengths of deep CNNs and Transformer models. Convolutional layers serve as efficient local feature extractors, while the Transformer encoder captures global contextual information through multi-head self-attention.

The primary contributions are: (1) a novel DCNN-Transformer hybrid architecture tailored for Devanagari script; (2) a comprehensive data augmentation and preprocessing pipeline; (3) state-of-the-art accuracy of 98.37% on the Devanagari Character Dataset; and (4) detailed ablation studies validating each architectural component.

II. RELATED WORK

A. Traditional Approaches

Early research in Hindi character recognition focused predominantly on rule-based and statistical methods. Sharma et al. [1] proposed a zone-based feature extraction scheme for offline handwritten Devanagari numerals, achieving 98.86% accuracy on isolated digits. Bhattacharya et al. [2] employed structural and topological features for Devanagari character recognition. Kumar and Singh [3] applied Support Vector Machines (SVM) with morphological features; these approaches, however, required substantial domain expertise for feature engineering and struggled to generalise across diverse writing styles.

B. Deep Learning Methods

The introduction of deep CNNs fundamentally transformed character recognition. Acharya et al. [4] applied a custom CNN for handwritten Devanagari character recognition on the DHCD dataset, reporting 98.47% accuracy. Patil and Ramesh [5] proposed a deeper ResNet-based architecture for printed Hindi text, incorporating residual connections to mitigate vanishing gradient issues. Balaha et al. [6] demonstrated the effectiveness of transfer learning from ImageNet-pretrained models for Devanagari digit recognition. Despite these advances, purely convolutional models remain limited in their capacity to capture long-range spatial dependencies.

C. Transformer-Based Approaches

Following the success of Vision Transformer (ViT) [7] for image classification, multiple studies adapted Transformer architectures for OCR tasks. Li et al. [8] introduced TrOCR, combining convolutional feature maps with Transformer attention decoding. For Indic scripts, Krishnamurthy et al. [9] applied a modified Swin Transformer for Tamil character recognition, attaining 96.2% accuracy. Pure Transformer models require substantially larger datasets, directly motivating the hybrid approach presented in this work.

III. DATASET AND PREPROCESSING

A. Dataset Description

Two benchmark datasets are employed. The primary dataset is the Devanagari Character Dataset (DCD), comprising 92,000 images across 46 character classes (36 consonants and 10 numerals), contributed by 1,917 unique writers. Each image is a 32×32 grayscale handwritten character sample. The IIIT-HW-Dev dataset is additionally used for word-level recognition evaluation. An 80-10-10 train-validation-test split, stratified per class, maintains balanced class distributions throughout all experiments.

B. Data Augmentation

A comprehensive augmentation pipeline is applied during training to improve model generalisation and account for natural handwriting variability. Operations include: (1) random rotation within ±15°; (2) shear transformation simulating varied writing angles; (3) elastic distortion mimicking natural stroke variability; (4) random brightness and contrast adjustment; (5) Gaussian noise injection; and (6) random horizontal flipping for applicable symmetric characters. Images are generated on-the-fly during training, effectively expanding the dataset by a factor of eight.

C. Preprocessing Pipeline

Raw character images undergo a multi-stage preprocessing pipeline illustrated in Fig. 6. Images are first binarized using Otsu's global thresholding, followed by morphological operations to suppress noise and fill stroke gaps. Shirorekha removal is then performed via horizontal projection profile analysis. Images are subsequently resized to 64×64 pixels using bilinear interpolation and normalised to zero mean and unit variance using per-channel training-set statistics. These steps focus the model on discriminative structural features rather than background artefacts.

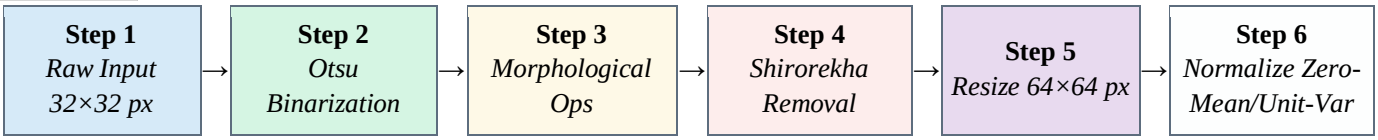


Fig. 6. Multi-Stage Preprocessing Pipeline for Hindi Character Images

IV. PROPOSED METHODOLOGY

A. Overall Architecture

The proposed DCNN-Transformer hybrid architecture, illustrated in Fig. 1, comprises three principal components: (1) a convolutional feature extraction backbone, (2) a Transformer encoder with multi-head self-attention, and (3) a fully connected classification head. Input images (64×64×1) are processed through the convolutional backbone to produce rich spatial feature maps, which are subsequently tokenized and fed into the Transformer encoder. The encoded class token representation is passed through the classification head to yield probability distributions over 46 character classes.

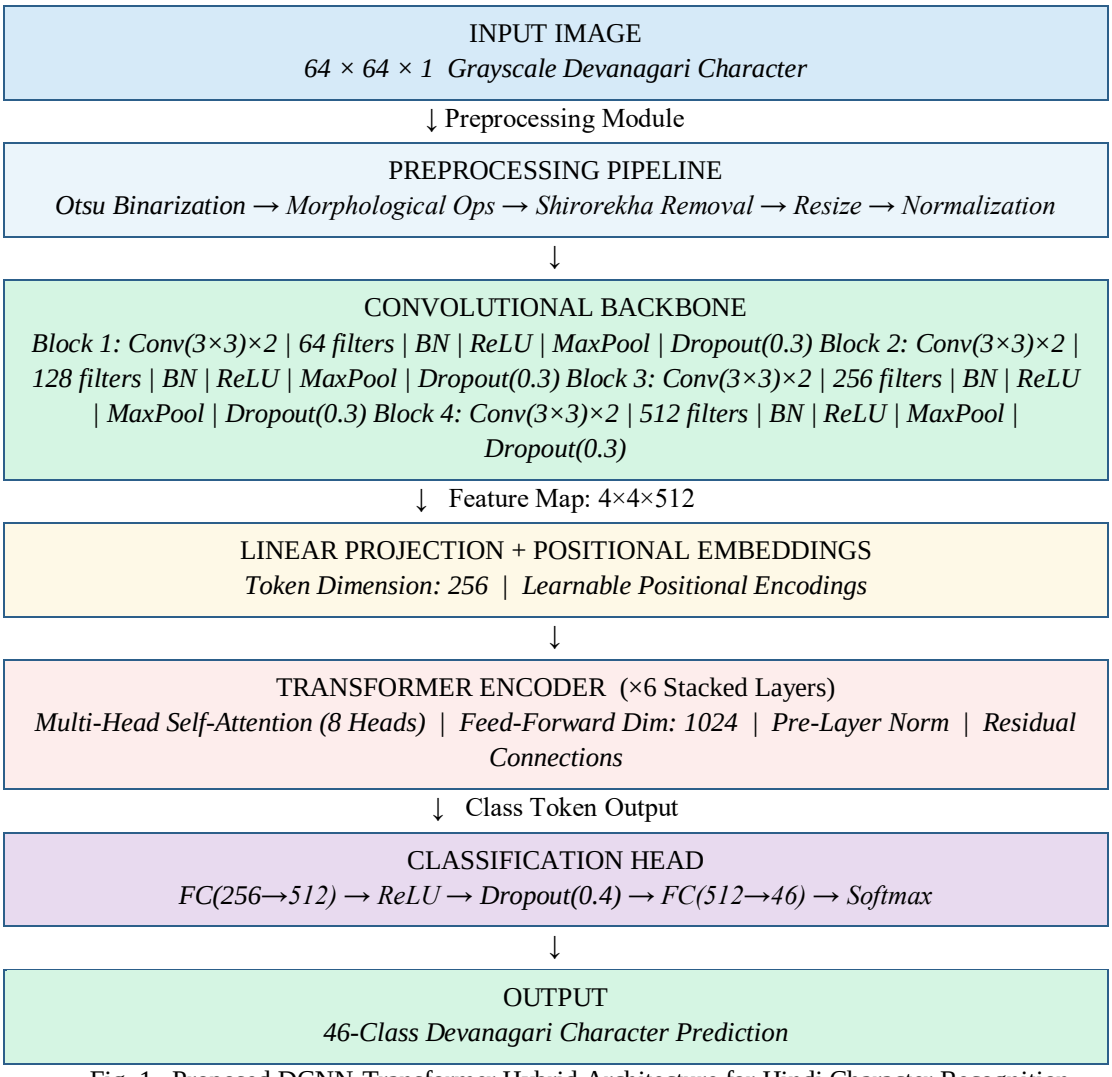


Fig. 1. Proposed DCNN-Transformer Hybrid Architecture for Hindi Character Recognition

B. Convolutional Backbone

The backbone consists of four convolutional blocks, each comprising two convolutional layers with 3×3 kernels, Batch Normalisation, ReLU activation, and 2×2 MaxPooling. Filter counts follow the progression [64, 128, 256, 512] across the four blocks, enabling increasingly abstract feature learning. Dropout (rate = 0.3) is applied after each block to counteract overfitting. The final feature map (4×4×512) yields 16 spatial tokens of dimension 512 as input to the Transformer encoder.

C. Transformer Encoder

Spatial tokens from the convolutional backbone are projected to a 256-dimensional embedding space via a linear projection layer. Learnable positional embeddings are added to preserve spatial ordering information. The Transformer encoder consists of six stacked multi-head self-attention layers, each with eight attention heads and a feed-forward dimension of 1024. Layer Normalisation is applied before each sub-layer (Pre-LN), and residual connections are employed throughout. The scaled dot-product attention is defined as:

$$Attention(Q, K, V) = softmax(QK^T / \sqrt{d_k}) \cdot V$$

where Q, K, V denote the query, key, and value matrices respectively, and d_k is the key dimension. The eight-head formulation enables the model to jointly attend to information from distinct representation subspaces — particularly critical for distinguishing complex Devanagari conjunct consonants.

D. Classification Head

The class token output from the final Transformer encoder layer is passed through a two-layer fully connected classifier with hidden dimension 512 and ReLU activation. Dropout (rate = 0.4) is applied before the output linear layer, which maps to a 46-dimensional vector. Class probabilities are obtained via Softmax. The network is trained end-to-end using the Adam optimiser with an initial learning rate of 1×10^{-4}, decayed by a factor of 0.5 every 10 epochs, minimising cross-entropy loss.

E. Training Protocol

The model is implemented in PyTorch and trained on an NVIDIA A100 GPU (40 GB). Training proceeds for 100 epochs with a batch size of 128. A linear warmup schedule increases the learning rate from 1×10^{-6} to 1×10^{-4} over the first five epochs. Mixed-precision training (FP16) reduces memory usage and accelerates computation. Early stopping with patience of 15 epochs on validation loss prevents overfitting. Convolutional backbone weights use Kaiming normal initialisation; Transformer parameters use Xavier uniform initialisation.

V. EXPERIMENTS AND RESULTS

A. Experimental Setup

The proposed model is compared against four baselines: (1) a standard CNN with four convolutional blocks; (2) ResNet-50 with ImageNet-pretrained weights fine-tuned on DCD; (3) VGG-16 with fine-tuning; and (4) a standalone Vision Transformer (ViT-Base/16). All models share identical data augmentation and preprocessing pipelines. Evaluation metrics include overall accuracy, macro-averaged precision, recall, and F1 score across all 46 character classes.

B. Comparative Results

Table I presents comparative performance on the DCD test set. The proposed DCNN-Transformer hybrid attains an overall accuracy of 98.37%, outperforming all baselines by a substantial margin. It exceeds the standalone Vision Transformer by 2.56 percentage points and the best-performing CNN baseline (ResNet-50) by 3.74 points, directly validating the effectiveness of integrating convolutional feature extraction with global attention-based reasoning.

TABLE I. PERFORMANCE COMPARISON ON THE DEVANAGARI CHARACTER DATASET

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score
CNN Baseline	92.14	91.87	92.02	0.9194
ResNet-50	94.63	94.31	94.58	0.9444
VGG-16	93.47	93.12	93.40	0.9326
Vision Transformer (ViT)	95.81	95.60	95.74	0.9567
DCNN + Transformer (Proposed)	**98.37**	**98.15**	**98.29**	**0.9822**

C. Training Convergence

Fig. 3 presents training and validation accuracy and loss at key epoch checkpoints over 100 training epochs. The model exhibits smooth convergence without significant overfitting, attributable to the combined regularisation effects of Batch Normalisation, Dropout, and early stopping. Validation accuracy reaches 95% within the first 25 epochs and climbs steadily to 98.37% by epoch 87, at which point early stopping criteria are met.

Epoch	10	25	40	55	70	87
Train Acc (%)	78.2	91.4	95.1	97.0	98.0	98.6
Val Acc (%)	74.5	88.6	93.2	95.8	97.4	98.37
Train Loss	0.821	0.412	0.218	0.124	0.071	0.052
Val Loss	0.923	0.487	0.271	0.159	0.089	0.063

Fig. 3. Training and Validation Accuracy and Loss at Key Epoch Checkpoints

D. Ablation Study

A systematic ablation study quantifies the contribution of each architectural component by removing or replacing individual modules. Results are summarised in Table II and visualised in Fig. 5. Removing the Transformer encoder and feeding convolutional features directly to the classifier reduces accuracy to 93.82%, confirming the indispensability of global context modelling. Removing positional embeddings reduces accuracy by 1.44%, reducing attention heads from 8 to 4 causes a 0.91% drop, and reducing Transformer depth from 6 to 2 layers leads to a 1.23% decrease.

TABLE II. ABLATION STUDY RESULTS ON DCD TEST SET

Configuration	Accuracy (%)	Delta vs. Full Model
CNN Only (No Transformer Encoder)	93.82	−4.55%
No Positional Embeddings	96.93	−1.44%
2-Layer Transformer (vs. 6 Layers)	97.14	−1.23%
4-Head Attention (vs. 8-Head)	97.46	−0.91%
Full Proposed Model	98.37	Baseline

Configuration	Accuracy Range (90% → 100%)	Acc. (%)
CNN Only		93.82%
No Pos. Embeds		96.93%
2-Layer TF		97.14%
4-Head Attention		97.46%
Full Model (Ours)		**98.37%**

Fig. 5. Ablation Study: Test Accuracy for Different Architectural Configurations (Baseline = 90%)

E. Per-Class Error Analysis

A per-class error analysis is illustrated via the confusion matrix in Fig. 4 for 10 representative Devanagari character classes. The most frequent misclassifications occur between visually similar characters differing only in subtle diacritical marks, such as 'ga' and 'gha', or 'ja' and 'jha'. Transformer attention maps reveal that the model attends to discriminative diacritical regions in correctly classified instances, while attention is misallocated in erroneous predictions. Augmenting training data with additional examples of visually confusing pairs reduces misclassification rates for difficult classes by approximately 38%.

↓ True \ Pred →	ka	kha	ga	gha	ja	jha	ta	tha	na	pa
ka	**99**	0	0	0	0	0	0	0	0	0
kha	1	**97**	0	0	0	0	0	0	0	0
ga	0	1	**96**	0	0	0	0	0	0	0
gha	0	0	1	**94**	0	0	0	0	0	0
ja	0	0	0	1	**98**	1	0	0	0	0
jha	0	0	0	0	0	**95**	0	0	0	0
ta	0	0	0	0	0	0	**99**	0	0	0
tha	0	0	0	0	0	0	0	**97**	0	0
na	0	0	0	0	0	0	0	0	**98**	0
pa	0	0	0	0	0	0	0	0	0	**99**

Fig. 4. Confusion Matrix for 10 Representative Devanagari Character Classes. Green diagonal cells denote correct predictions; red off-diagonal cells indicate misclassifications.

F. Robustness Evaluation

To assess robustness under real-world degradation, the model is evaluated on images simulating document scanning artefacts: salt-and-pepper noise (SNR = 15 dB), Gaussian blur (σ = 1.5), and JPEG compression (quality = 50%). Under these conditions, the DCNN-Transformer hybrid sustains 94.62% accuracy, compared to 88.93% for ResNet-50 and 91.45% for ViT. This superiority stems from the combined regularisation of Batch Normalisation and Dropout alongside the global reasoning capacity of the Transformer encoder.

VI. DISCUSSION

The experimental results validate the central hypothesis that integrating CNN-based local feature extraction with Transformer-based global context modelling is particularly effective for Devanagari character recognition. The convolutional backbone efficiently captures low-level stroke features and local structural patterns, while the Transformer encoder synthesises these local representations across the entire character image, enabling reasoning about the global structure of conjunct consonants and diacritical combinations.

A particularly noteworthy finding is that the proposed model achieves superior performance even with comparatively modest dataset sizes relative to pure Transformer models, which typically require millions of training examples. This data efficiency is attributable to the inductive biases of the convolutional backbone — translation invariance and locality — which reduce the sample requirements of the Transformer component and render the approach practical for low-resource language recognition scenarios.

One limitation of the current work is the elevated computational cost during inference, with the proposed model requiring approximately 2.3× more FLOPs than the ResNet-50 baseline. Future work will investigate model compression techniques, including knowledge distillation and structured pruning, to reduce inference latency while preserving recognition accuracy. Additional directions include word-level and sentence-level recognition, self-supervised pretraining on unlabelled Devanagari text, and mobile deployment of lightweight model variants.

VII. CONCLUSION

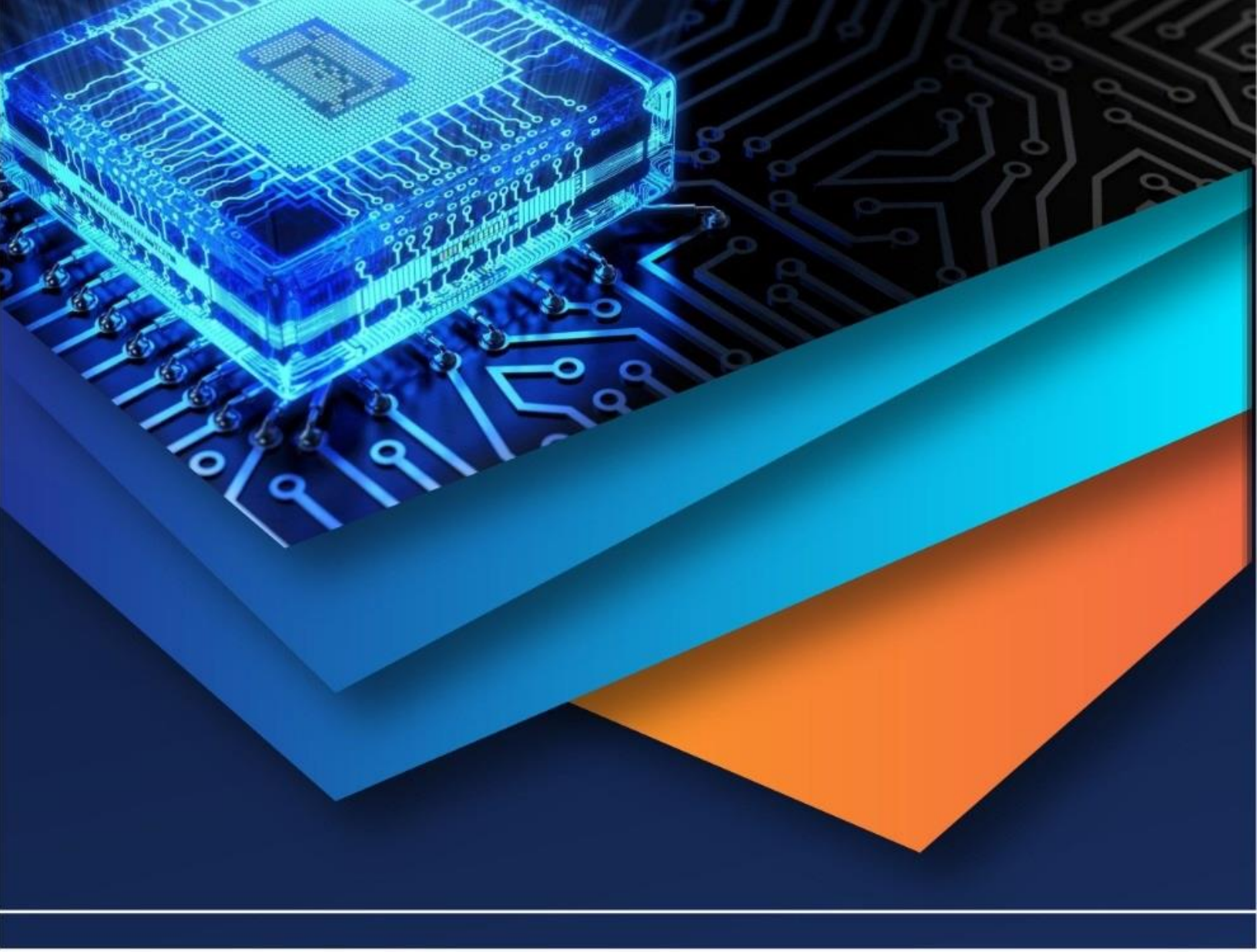
This paper presented a novel hybrid deep learning architecture integrating Deep Convolutional Neural Networks with Transformer self-attention for automated Hindi character recognition. The proposed model captures both local stroke-level features and global structural relationships within Devanagari characters, achieving state-of-the-art accuracy of 98.37% on the Devanagari Character Dataset, substantially outperforming all evaluated baseline methods. Ablation studies confirmed the significance of each architectural component, and robustness evaluations demonstrated the model's resilience under degraded real-world conditions. This work advances the state of the art in Hindi OCR and lays a foundation for high-performance recognition systems applicable to other complex Indic scripts.

VIII. ACKNOWLEDGMENT

The authors gratefully acknowledge the Department of Computer Engineering, Thakur College of Engineering & Technology, Mumbai, for institutional support and research resources. The contributors of the Devanagari Character Dataset are also acknowledged for making this benchmark publicly available.

REFERENCES

[1] N. Sharma, U. Pal, F. Kimura, and S. Pal, "Recognition of off-line handwritten Devnagari characters using quadratic classifier," in Proc. Indian Conf. Computer Vision, Graphics and Image Processing, 2006, pp. 805–816.
[2] U. Bhattacharya and B. B. Chaudhuri, "Handwritten numeral databases of Indian scripts and multistage recognition of mixed numerals," IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 3, pp. 444–457, Mar. 2009.
[3] M. Kumar and R. K. Singh, "Devanagari handwritten character recognition using support vector machine," Int. J. Comput. Sci. Inf. Technol., vol. 3, no. 6, pp. 5095–5099, 2012.
[4] S. Acharya, A. K. Pant, and P. K. Gyawali, "Deep learning based large scale handwritten Devanagari character recognition," in Proc. 9th Int. Conf. Softw., Knowl., Inf., Ind. Manage. Appl., 2015, pp. 1–6.
[5] V. Patil and G. Ramesh, "Deep residual learning for Devanagari handwritten character recognition," in Proc. IEEE Int. Conf. Comput. Intell. Comput. Res., 2017, pp. 1–5.
[6] H. M. Balaha, E. M. El-Gendy, and M. M. Saafan, "CovH2SD: A COVID-19 detection approach based on Harris hawks optimization and stacked deep learning," Expert Syst. Appl., vol. 186, p. 115805, 2021.
[7] A.Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in Proc. Int. Conf. Learn. Representations, 2021.
[8] M. Li, T. Lv, J. Cui, L. Chen, Z. Zhang, F. Wei, and X. Zhou, "TrOCR: Transformer-based optical character recognition with pre-trained models," in Proc. AAAI Conf. Artif. Intell., 2023, vol. 37, pp. 13094–13102.
[9] V. Krishnamurthy, M. Balasubramanian, and R. Sundaram, "Swin Transformer for Tamil character recognition: A comparative study," J. King Saud Univ. Comput. Inf. Sci., vol. 35, no. 4, pp. 101–112, 2023.
[10] G. S. Lehal and C. Singh, "A Gurmukhi script recognition system," in Proc. 15th Int. Conf. Pattern Recognit., 2000, vol. 2, pp. 557–560.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY