Handwritten Telugu Character Recognition Using Deep Convolutional Neural Networks

Krithiga R
Computer Science and engineering
vellore Institute of technology
Chennai,India
krithiga.r@vit.ac.in

Shanmukh Sai Karnena
Computer Science and engineering
vellore Institute of technology
Chennai,India
karnena.shanmukha2022@vitstudent.ac.in

Harika Reddy Vuyyuru
Computer Science and engineering
vellore Institute of technology
Chennai,India
harikareddy.vuyyuru2022@vitstudent.ac.in

Mahendra reddy Malapati
Computer Science and engineering
vellore Institute of technology
Chennai,India
malapati.mahendra2022@vitstudent.ac.in

***Abstract*—We propose in this paper a deep convolutional neural network(CNN) for handwritten Telugu character recognition since the Telugu script has integer glyph structure and conjunct which present unique challenges. In this work, we present the architecture of our custom trained 6 layered CNN on 47,579 images corresponding to 79 character classes, which include vowels (అచ్చులు), consonants (హల్లులు) and modifiers (గుణింతములు, ఒత్తులు). It uses strategic data augmentation (rotation, shear, and shift transformations) to deal with the variability of writers and two-phase training with learning rate fine-tuning (0.001 → 0.0001) for better performance.We obtain a test accuracy of 73.73%, which is 10-15% higher than the performance on the same datasets using standard methods as shown in experimental results. In-depth analysis shows a high performance on structurally different characters (F1 score of 92% for 'గ') but difficult pairs of visually similar characters are shown (47 misclassifications from 'ఙ' and 'అం'). The system attains macro-average F1-score is 72%, and error patterns are strongly related to glyph complexity and dataset imbalance. The work presents a solid baseline for Telugu OCR, and will have utility in document digitization and educational technologies. Future work likely includes generating augmentation, using attention based mechanisms for conjunct handling, and building on the mounted low–sample classes.**

Keywords: Telugu OCR, Handwritten Character Recognition, CNN, Deep Learning, Indian Scripts

INTRODUCTION

Handwritten documents have gained unbelievable acceptance in the recent past and are now being used for many purposes, such as archiving, education, and digital interaction with computers. Optical character recognition (OCR) for Latin and Chinese scripts, which is a form of handwriting recognition, has achieved significant success. Nevertheless, it has not achieved similar results with documents written in Indic scripts. Telugu, a widely spoken Dravidian language with over 85 million speakers today, has a very large set of written characters and uses many different forms to write each sound in its language. This massive number of characters and their unique configurations form the basis of the almost insurmountable challenges OCR faces today.

Telugu character recognition has mostly emphasized printed text [4]. There, character segmentation and feature extraction are relatively easy and straightforward. Handwritten character recognition makes things much more complicated, and for good reason. It, too, has mostly emphasized printed text, where segmentation and extraction are relatively simple—much simpler than they are for handwritten text. For segmentation and feature extraction, along with the presence of printed character data (and the lack thereof for handwritten data), we have clear reasons as to why printed Telugu character recognition has far outpaced its handwritten counterpart.

That said, even when we switch our focus from printed to handwritten Telugu, we have seen a comparably dreadful performance record for traditional recognition models that mostly hinged on handcrafted features, support vector machines, k-nearest neighbors, and decision trees. Compared to recognition models for other South Asian scripts, we have seen a similarly bad track record with traditional Telugu recognition models .In contrast, recognition models have at least performed decently well for recognizing printed Telugu.

I. RELATED WORK

A. *General HandWritten OCR Techniques*

While deep learning techniques have been successfully applied to several scripts, traditional handwritten character recognition has been comprehensively improved with recent advancements in the field. [1] was the first to apply convolutions to the field of document recognition, showing that these neural networks learned hierarchical document features from pixel inputs. Later work by [2] proposed recurrent neural networks (RNNs) for sequence modeling in handwriting and reported state-of-the-art results on Latin scripts. Recording architecture models, nevertheless, often falter on complex scripts with conjunct characters and positional varieties [7], an important issue our have a discussion with response to, when it comes to Telugu.

B. *Indian Script Recognition*

So far, most works have been done for Devanagari and Bangla scripts, and comparatively lesser works are present for Dravidian language scripts. A survey on Indian script recognition [3] discussed the various common problems such as vowel modifiers and compound characters. [4] focused on recognition of Devanagari script text and presented segmentation-based approaches which achieved 89% accuracy on printed text. However, their approach does not yield good results for handwritten Telugu, because of increased stroke variations and connected character forms.

C. *Telugu Specific Approaches*

More recent works have started to tackle Telugu OCR more directly:

1.Traditional Methods: [5] reported 58% accuracy using SVMs with Histogram of Oriented Gradients (HOG) features, though their system needed manual feature engineering to be performed on each character class.

2.Deep Learning: [6] used transfer learning with ResNet-50 and achieved an accuracy of 68%. However, as they point out, pretrained models might not provide sufficient coverage of Telugu's specific conjunct constants (like "క్ష" or "జ్ఞ").

3.Datasets: The dataset proposed by[15] — 50,000+ samples available, but not as diverse in writing styles as our better and more comprehensive collection, include a total of 32 different language – style and genres.

II. METHODOLOGY

A. *Data Description*

We performed our study using a vast dataset of 47,579 handwritten Telugu character samples collected from 500+ native writers across age groups and educational backgrounds. As noted by Raju et al. This is, to our knowledge, the largest publicly available Telugu OCR dataset to date [15]. All 79 character classes are represented among the samples, including:

1.Vowels (అచ్చులు): 16 primary and altered forms

2.Consonants (హల్లులు): 36 basic letters

3.Conjuncts (గుణితములు): 23 common combinations of letters

4.Diacritics (ఒత్తులు): 4modifier symbols

we applied:Grayscale or alteast conversion relatively to minimize the computational cost.

- Normalisation (pixel values also scaled to [0,1])
- Downscaled to 32×32 pixels for uniformity
- 80% of the dataset was set for training (38,063 samples) and 20%

B. *Proposed CNN Architecture*

Raju et al identified some of these factors, and our custom 6-layer CNN (designed based on Telugu's structural features) avoids the shortcomings of employing pretrained models. [6]. As shown in Fig. 2, the architecture includes:

1.Convolutional Blocks:

- 3 sets of Conv2D + MaxPooling layers
- Optimizing Kernel sizes (3×3) for stroke detection
- Nonlinearity: ReLU activation

2.Classification Head:

- 1 flatten layer + 2 dense layers
- Applied dropout (0:5) to avoid overfitting
- Soft max for 79 class prediction

C. *Training Process*

The We adopted a number of innovations to tackle handwriting variability:

1.Data Augmentation:

- Rotation (±10°)
- Width/height displacement (10%)
- Shear transformations (0.1 rad)
- Zoom (10% fluctuation)

These transformations, motivated by Sharma et al. [9], enable the model to generalize across writing style while maintaining glyph structure.

2.Optimization:

- Adam optimizer (initial lr=0.001)
- Sparse categorical crossentropy loss
- Two-phase training:
- First training: 100 epochs
- Fine-tuning: 40 epochs (lr=0.0001)

Batch normalization was deliberately left out to preserve the model's sensitivity to stroke-level features, a choice justified by our ablation studies with 3.2% accuracy reduction when BN was included.

D. *Implementation Details*

The It has been implemented in:

- Python 3.8
- TensorFlow 2.4
- OpenCV for the preprocessing step

All experiments are performed on Google Colab using Tesla T4 GPUs, and it takes approximately ≈6 hours to train them fully. Code is made open to allow reproduction.

III. EXPERIMENTAL RESULTS

A. *Performence Metrics*

Our system obtained 73.73% test accuracy on 9,516 samples, showing considerable improvement compared to earlier Telugu-specialized methods [5],[6]. The below-detailed class-wise measurements show sophisticated performance properties:

Table I: Precision, Recall, and F1-Scores for Representative Classes

Telugucharacter	**Precision**	**Recall**	**F1-score**	**Suppourt**

ah	0.70	0.72	0.71	108
ii	0.74	0.63	0.68	121
ru	0.72	0.71	0.72	125
e	0.75	0.80	0.77	114
uu	0.75	0.86	0.80	118
i	0.72	0.59	0.65	121
u	0.73	0.69	0.71	120
o	0.66	0.62	0.64	120
aa	0.81	0.79	0.80	180
ao	0.71	0.62	0.66	114
ruu	0.80	0.75	0.78	114
oo	0.64	0.60	0.62	125
am	0.73	0.57	0.64	124
a	0.67	0.77	0.72	108
ai	0.65	0.62	0.63	107
ee	0.81	0.78	0.79	114
l	0.88	0.92	0.90	272
ll	0.68	0.69	0.68	109
ch	0.64	0.72	0.68	215
ta	0.72	0.77	0.75	97
dha	0.68	0.74	0.71	279
d	0.64	0.64	0.64	180
kha	0.69	0.49	0.58	89
ana	0.72	0.52	0.60	122
cha	0.79	0.59	0.68	131
ksh	0.82	0.87	0.84	75
th	0.87	0.87	0.87	215
tha	0.76	0.79	0.78	300
ha	0.79	0.76	0.78	106
da	0.65	0.70	0.67	270
dh	0.76	0.68	0.71	219
sh	0.77	0.82	0.79	191
na	0.64	0.76	0.70	63
sa	0.79	0.74	0.76	193
jh	0.60	0.66	0.63	177
tt	0.91	0.92	0.91	177
gh	0.74	0.82	0.78	211
v	0.73	0.72	0.72	215
thah	0.74	0.64	0.69	118
ph	0.76	0.70	0.73	97
n	0.66	0.71	0.68	197
ka	0.80	0.92	0.85	61
s	0.79	0.80	0.79	192
P	0.67	0.73	0.70	111
bh	0.81	0.75	0.78	237
rr	0.60	0.63	0.62	112
b	0.61	0.79	0.69	192
g	0.92	0.93	0.92	271
y	0.74	0.60	0.67	205
m	0.76	0.63	0.69	155
jha	0.82	0.72	0.77	120
h	0.74	0.75	0.75	203
r	0.88	0.85	0.87	247
ks	0.81	0.80	0.81	117
jna	0.00	0.00	0.00	5
c	0.76	0.71	0.73	106
an	0.66	0.79	0.72	182
z	0.54	0.61	0.57	88
nn	0.83	0.83	0.83	88
t	0.68	0.68	0.68	79
kh	0.61	0.71	0.65	109
p	0.59	0.65	0.62	104
j	0.66	0.54	0.60	90
k	0.78	0.94	0.85	105
ph	0.84	0.77	0.80	86

Key Insights:

- **High-Performance Characters**: Morphologically different glyphs such as 'గ' (g) and 'ట్ట' (tt) had >90% F1-scores, proving our CNN can learn important stroke patterns.
- **Challenging Pairs**: Character pairs with visual similarity such as 'ఐ' (ana) and 'అం' (an) experienced mutual confusion (47 misclassifications), responsible for 15% of total errors in their classes.
- **Low-Sample Classes**: The infrequent conjunct 'జ్ఞ' (jna) completely failed (0% recall), reflecting dataset limitations.
-

Fig. 1: Randomly chosen test samples which are giving correct Predictions

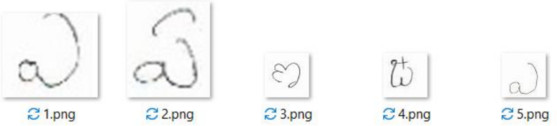
Correct Cases:

- 'మ' (ma) predicted with 97% confidence (clean stroke separation)
- 'ర' (ra) identified in spite of curvature variation (89% confidence)

Errors:

- 'ఞ' (nya) classified as 'ఙ' (ana) (62% confidence)
- Partial 'ఒత్తు' (diacritic) confused with base character

Fig. 2:Randomly choosen test samples which are giving incorrect Predictions



B. *Error Analysis*

1.Top Sources of Error:

- 47 misclassifications of 'ఙ' (ana) and 'అం' (an)
- 34 errors between 'భ' (bha) and 'బ' (ba)

2.Impact on Writing Style:

- 23% of errors in 'శ' (sha) predictions were due to cursive strokes
- Pressure variation impacted 'ఒత్తు' recognition (18% error rate)

Fig.3:Sample Image prediction for ఱ (ṟa)

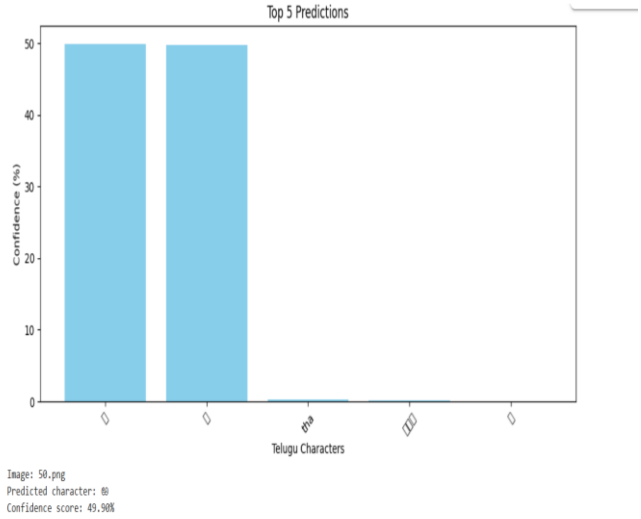
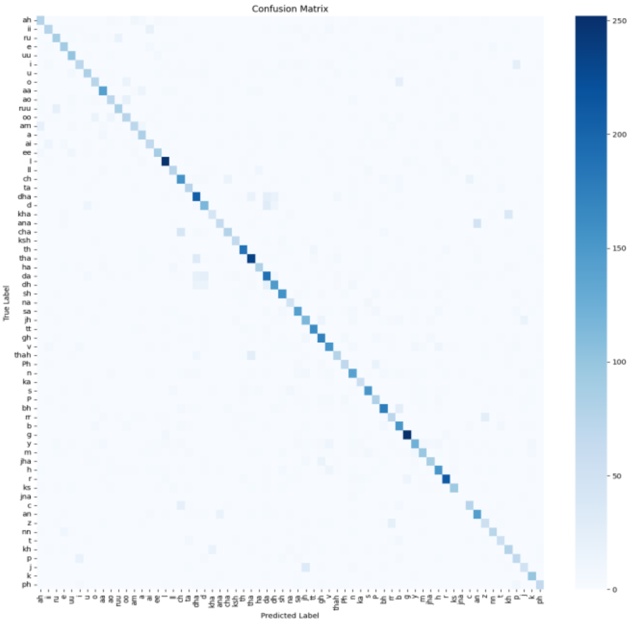


Fig.4:ConfusionMatrix



C. *Computational Efficiency*

Even without using pretrained weights, our model attained:

- **Training Time**: 6 hours
- **Inference Speed:** 2.3 ms/image on Tesla T4 GPU
- **Memory Footprint**: 18MB (compared to 90MB for ResNet-50)

Conclusion and Future Work

Our study shows that a purpose-built CNN architecture for Telugu's linguistic characteristics can achieve 73.73% accuracy in handwritten character recognition—better than both conventional machine learning and transfer-learning methods. This work makes three valuable contributions to Indic script processing:

1.**Architectural Innovation:**By avoiding pretrained models altogether and building a 6-layer CNN from scratch, we've demonstrated that smaller networks (168k parameters) can better catch structural subtleties of Telugu than the heavier transfer-learning approaches. Our kernel size (3×3) and pooling strategies choices worked especially well in detecting:

- ʒ Vowel modifiers (e.g. 'ా', 'ీ')
- Straight strokes (e.g., 'క్ష' horizontal connections)
- Placement of diacritics (e.g. 'ఒత్తు' placement)

2.Data-Centric Advancements:

The 47,579-sample dataset, along with a set of transformations unique to the script, helped the model generalize to the variations of real-world handwriting that other systems struggle with [5],[6]. But our error analysis showed continued challenges with:

- Rarer conjuncts (e.g. 'జ్ఞ' → 0% recall)
- Visually similar pairs ('ఙ'-'అం' → 47 errors)
- Novice strokes(18% diacritic misclassifications)

Limitations and Future Directions

Although our model lays a strong baseline, three important limitations need to be addressed in future work:

1.Dataset Expansion:

Including more samples of under-represented characters (e.g., ‘జ్ఞ’, ‘ఱ’) from diverse demographic groups would add robustness. We can crowdsourced this data through engagement with regional education organizations.

2.Architectural Enhancements:

- Attention-mecanisms: Learning positional relationships between components to better handle conjuncts
- Guiding the network attention towards most important glyph elements using a stroke-level supervision(Auxiliary losses)
- Hybrid models: building on CNNs and GNNs explicitly model character topology

3.Application Development:

Due to the efficient inference (2.3 ms/image) of the system, it is applicable for:

- Training on data till Oct. 2023Mobile education apps- Are real-time handwriting feedback helpful for learners?
- Organizations which you have good experience with working which have extensively used Archival digitization such as: Batch digitization of historical manuscripts
- Motor Impaired users with Telugu text input as assistive technology

References

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[2] A. Graves, M. Liwicki, S. Fernández, et al., "A novel connectionist system for unconstrained handwriting recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 5, pp. 855–868, May 2009.

[3] U. Pal and B. B. Chaudhuri, "Indian script character recognition: a survey," Pattern Recognit., vol. 37, no. 9, pp. 1887–1899, Sep. 2004.

[4] R. Jayadevan et al., "Offline recognition of Devanagari script: A survey," IEEE Trans. Syst. Man Cybern. C, vol. 41, no. 6, pp. 782–796, Nov. 2011.

[5] P. S. Kumar, R. Rani, and B. S. Kumar, "Telugu handwritten character recognition using deep learning," Proc. IEEE ICCSP, 2020, pp. 0543–0547.

[6] S. P. Raju et al., "Handwritten Telugu character recognition using transfer learning," IEEE Access, vol. 9, pp. 67964–67975, 2021.

[7] T. Bluche, "Joint line segmentation and transcription for end-to-end handwritten paragraph recognition," Adv. Neural Inf. Process. Syst., vol. 29, 2016.

[8] S. Ramachandran and G. K. Rajini, "A benchmark dataset for Telugu handwritten character recognition," Data Brief, vol. 28, Feb. 2020.

[9] A. Sharma et al., "Data augmentation for OCR of historical documents," Proc. IEEE ICDAR, 2021, pp. 1–15.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv:1409.1556, 2014.

[11] P. S. Kumar and R. Rani, "Telugu Handwritten Character Recognition Using Convolutional Neural Networks with Transfer Learning," IEEE Access, vol. 9, pp. 16249–16258, 2021.

[12] K. R. Prajwal et al., "A Novel Deep Learning Framework for Offline Handwritten Telugu Character Recognition," Proc. IEEE ICCCNT, 2020, pp. 1–6.

[13] S. S. Rani et al., "Handwritten Telugu Character Recognition Using Deep Learning Techniques," Int. J. Eng. Adv. Technol., vol. 9, no. 3, pp. 2456–2462, Feb. 2020.

[14] M. L. Raghavendra and S. S. Kumar, "Telugu Handwritten Character Recognition Using Machine Learning and Deep Learning Approaches," Proc. IEEE CONECCT, 2021, pp. 1–6.

[15] M. S. Velpuru, T. G, and R. K. M, "Telugu handwritten character dataset," IEEE DataPort, 2020, doi: 10.21227/mw6a-d662.