

Puzzle optimization algorithm with capsule networks based Telugu character recognition model

Josyula Siva Phaniram, Mukkamalla Babu Reddy

Department of Computer Science, Krishna University, Machilipatnam, India

Article Info

Article history:

Received May 14, 2022

Revised Jun 18, 2022

Accepted Jul 4, 2022

Keywords:

Data collection

Deep learning

Normalization

Puzzle optimization algorithm

pre-processing

Telugu character recognition

capsule network

ABSTRACT

Recently, Telugu character recognition (TCR) becomes a hot research topic because of drastic increase in technological advancements such as multimedia, and smartphones. Though numerous works have been concentrated on offline TCR models, it is still needed to develop automated and intelligent online TCR models. This paper presents a novel puzzle optimization algorithm with capsule networks-based Telugu character recognition (POACN-TCR) model. The presented POACN-TCR model intends to effectively identify and recognize distinct Telugu characters online. To accomplish this, the POACN-TCR model primarily undergoes pre-processing in different ways such as normalization, smoothing, and interpolation. In addition, the POACN-TCR model designs an effective capsule network (CapsNet) model to generate feature vectors and hyperparameter optimization takes place using puzzle optimization algorithm (POA). Finally, the C4.5 decision tree classifier is utilized for the effectual recognition of Telugu characters. The utilization of POA for hyperparameter optimization of the CapsNet model helps in achieving improved recognition performance. For ensuring the enhanced outcomes of the POACN-TCR model, a wide-ranging experimental analysis is performed and the outcomes pointed out the betterment of the POACN-TCR model on the recent approaches.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Josyula Siva Phaniram

Department of Computer Science, Krishna University

Machilipatnam, Andhra Pradesh, India

Email: phaniram.research@gmail.com

1. INTRODUCTION

Handwritten character detection was assumed that one of the appeals and stimulates research areas from the field of computer vision and pattern detection [1], [2]. Due to the major component of differences in cursive text and writing pattern, and the comparison of several characters from the procedure, recognition research was challenging and time utilized [3]. The online character detection was somewhat easy due to the temporal-based character properties such as procedure, count of distances, stroke, and writing direction. The offline character identification performance was complex due to their difference in fonts and writers [4], [5].

Telugu is the ancient and old language comprising Consonants-36, Vowels-16, and Guninthalu (Character generated by the combination of Telugu Vowels and Consonants) 560, around 612 characters [6]. Due to their multi classification problems, it could be difficult to recognize the handwritten Telugu characters, particularly, in Telugu Guninthalu, as all the characters are closely compared with one another. Many scientists are tried in identifying Indian languages particularly Telugu was explained from the survey to detect compound characters with template equivalent and several other approaches [7], [8]. Many machine learning (ML)

approaches were utilized to Pashto handwritten character recognition [9]. The power of deep neural networks (DNN) is for providing effectual implementation with no assuming the basic rule of deep element is linear [10].

Bhat *et al.* [11] present a method for the detection and classification of ancient Tigalari characters in the handwritten text. Tigalari is extremely utilized from coastal Karnataka and Kerala to document Malayalam, Sanskrit, and Tulu languages. This technique contains the design of database, proposal of deep convolutional neural network (DCNN) based framework for classifying the text, trained the method with data and identifying text utilizing test set. Prathima and Muppalaneni [12] refer to the handwritten Telugu vowels detection by utilizing a CNN method. The dataset is pre-processing and also feature extracting utilizing DNN method to train.

Jaisswal *et al.* [13] generate a deep learning (DL) technique which is classifying up to 12 distinct languages, at document level and demonstrates the resultant dependence upon the outcome by executing line-level segmentation, then word-level segmentation, and lastly with character detection of the documents. Guha *et al.* [14] different CNN approach on publicly accessible hand-written Devanagari character and numeral dataset. It can mostly concentrate on comparative works by assuming memory utilization, training time, and trainable parameters. It can be planned and presented DevNet, an altered CNN structure that creates feasible outcomes since memory space and computation complexity are most important issues in this method [15]-[19].

This paper presents a novel puzzle optimization algorithm with capsule networks-based Telugu character recognition (POACN-TCR) model. The presented POACN-TCR model undergoes pre-processing in different ways such as normalization, smoothing, and interpolation. In addition, the POACN-TCR model designs an effective capsule network (CapsNet) model to generate feature vectors and hyperparameter optimization takes place using POA. Finally, the C4.5 decision tree classifier is utilized for the effectual recognition of Telugu characters. For ensuring the enhanced outcomes of the POACN-TCR model, a wide-ranging experimental analysis is performed.

2. THE PROPOSED MODEL

In this study, a novel POACN-TCR model has been developed to effectively identify and recognize distinct Telugu characters online. Followed by, the POACN-TCR model designing an effective CapsNet model to generate feature vectors and hyperparameter optimization takes place using POA. Finally, the C4.5 decision tree classifier is utilized for the effectual recognition of Telugu characters. Figure 1 illustrates the overall process of POACN-TCR technique.

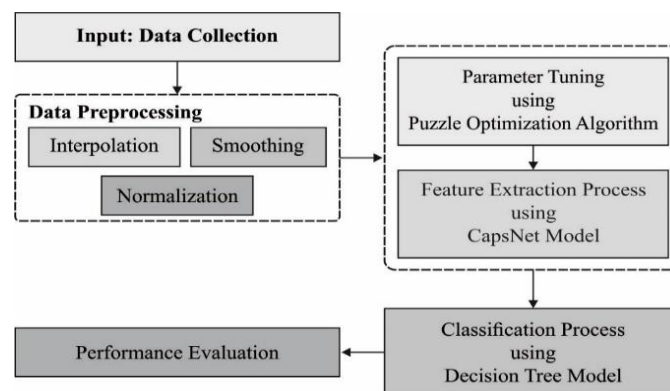


Figure 1. Overall process of POACN-TCR technique

2.1. Data pre-processing

The POACN-TCR model primarily undergoes pre-processing in different ways such as normalization, smoothing, and interpolation. Data preprocessing, a component of data preparation, describes any type of processing performed on raw data to prepare it for another data processing procedure. It has traditionally been an important preliminary step for the data mining process. More recently, data preprocessing techniques have been adapted for training machine learning models and artificial intelligence (AI) models and for running inferences against them. Data preprocessing transforms the data into a format that is more easily and effectively processed in data mining, machine learning and other data science tasks. The techniques are generally used at the earliest stages of the machine learning and AI development pipeline to ensure accurate results.

Normalization: some characters are written by various strokes with any strokes expanded below or above an initial part of characters. An initial part of characters offers a measure of character size and line space employed by writers.

Smoothing: gaussian filter was executed to smoothen the stroke. A 1D Gaussian distribution was taken to account by the smoothing kernel which is primarily convolved with x- and y-axis. The smoothing was reached dependent upon the size of window functions and the smoothing kernel parameter utilized.

Interpolation: the final stage in pre-processed in which the smoothing stroke was included to offer a set amount of points, likewise spaced with curve length. The count of points was chosen based on the average amount of points to all the strokes from the provided data sets.

2.2. CapsNet feature extraction

Once data pre-processing is done, the next level is to extract feature vectors. CapsNets mentions a completely novel type of DL structure that attempts for conquering the limitation and disadvantages of CNN techniques as lose valued data and lack the precise model of entity at the time of max-pooling. The typical CapsNet is shallower, with 3 layers namely PrimaryCaps, DigitCaps, and Convld layers. The capsule-based representation of the group of hidden neurons, whereas probability and property of hidden features are captured. During this sample, the CapsNet is strong for affine conversion and minimum trained data. Furthermore, the CapsNet has resulted in particular developments connected to spatial hierarchies amongst features. The capsule signifies the group of neurons [20]. The activity of neurons encircled from a dynamic capsule refers the different features of particular entity. While the length of resultant vectors signifies the probability of existence, the resultant capsule was computed by nonlinear squashing function.

$$v_j = \frac{\|s_j\|^2}{\epsilon + \|s_j\|^2} \frac{s_j}{\|s_j\|} \tag{1}$$

In which v_j denotes the vector outcome of capsules j and s_j refers the entire input. The non-linear squashing function is activation function to make sure that the short vector obtains shrunk to approximately zero length and longer vector obtains shrunk to particular length assuming ϵ .

$$s_j = \sum_i c_{ij} W_{ij} u_i. \tag{2}$$

The entire inputs to capsule s_j was obtained by multiplying the outcome u_i of capsule utilizing a weighted matrix W_{ij} which signifies a weight amount on every predicted vector in the capsule in the under layer. At this point, c_{ij} signifies the coupling co-efficient for instance, determined as the iterative dynamic routing process.

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})}, \tag{3}$$

In which b_{ij} & b_{ik} implies the log prior probabilities amongst 2 coupled capsules. The entire length of resultant vectors signifies the predicted chance. A separate margin loss L_k per capsule, k was:

$$L_k = T_k \max(0, m^+ - \|v_k\|)^2 + \lambda(1 - T_k) \max(0, \|v_k\| - m^-)^2, \tag{4}$$

whereas $T_k = 1$, $m^+ = 0.9$, and $m^- = 0.1$ signifies the three free variables by default. λ allow the down weighted of loss and support for ensuring latter convergence.

2.3. Hyperparameter optimization

For optimally modifying the hyperparameters related to the CapsNet model, the puzzle optimization algorithm (POA) is applied. POA is presented and its arithmetical model is shown for resolving optimization problems [21]. A presented method is a population-based approach that is established according to puzzle game simulation. The population in POA is arithmetically modelled by a matrix that is shown in (5).

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N \times m} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,d} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \cdots & x_{i,d} & \cdots & x_{i,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,d} & \cdots & x_{N,m} \end{bmatrix}_{N \times m} \tag{5}$$

Now, X_i indicates the i 'th puzzle, X denotes the population of puzzles, m represents the number of problem variables, N represented the number of population of puzzles, and $x_{i,d}$ indicates the value of d 'th variable recommended by i 'th puzzle. Assume that every member of the population is a solution to the optimization issue, the value of objective function is estimated. Thus, equivalent to the number of population members, the objective function is estimated that the attained value for objective function is simulated.

$$F = \begin{bmatrix} f_1 \\ \vdots \\ f_i \\ \vdots \\ f_N \end{bmatrix} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N \times 1} \quad (6)$$

Now, F denotes the vector of attained value for objective function and f_i is represent value of objective function of i 'th puzzle. As per the comparison of value attained for the objective function, the member that provides the optimal value for the objective function is known as the optimal member of the population. It is shown in:

$$B = X_k, f_k = \min (F) \quad (7)$$

where B indicates the optimal member and X_k denotes the k 'th puzzle with minimal objective function equivalent to f_k . In the presented approach, population member is upgraded in two phases. Here, it is arithmetically expressed as:

$$GM_i = X_g, g \in \{1,2,3, \dots, N\} \quad (8)$$

$$dx_{i,d} = \begin{cases} (GM_{i,d} - I \times x_{i,d}), F_g < F_i \\ (x_{i,d} - I \times GM_{i,d}), else \end{cases} \quad (9)$$

$$I = \text{round}(1 + \text{rand}) \quad (10)$$

$$X_i^{\text{new}} = X_i + r \times dx_i \quad (11)$$

$$X_i = \begin{cases} X_i^{\text{new}}, F_i^{\text{new}} < F_i \\ X_i, else \end{cases} \quad (12)$$

Now, $GM_{i,d}$ represent d 'th dimension, F_g denotes value of objective function, GM_i denotes the guiding member of i 'th puzzle, $dx_{i,d}$ signifies the change of d 'th dimension of i 'th puzzle, I shows a random number that is 1 or 2, r denotes an arbitrary value within [0, 1] interval, X_i^{new} indicates the novel status of i 'th puzzle, and F_i^{new} denotes the value of objective function. Next, every member of the population upgrades their status with the help of puzzle pieces recommended by other members of the population. It can be arithmetically expressed in (13)-(15).

$$N_p = \text{round} \left(0.5 \times \left(1 - \frac{t}{T} \right) \times N \right) \quad (13)$$

$$x_{i,d_j}^{\text{new}} = x_{h,d_j} \begin{cases} h \in \{1,2,3, \dots, N\} \\ j \in \{1,2,3, \dots, N_p\} \\ d_j \in \{1,2,3, \dots, m\} \end{cases} \quad (14)$$

$$X_i = \begin{cases} X_i^{\text{new}}, F_i^{\text{new}} < F_i \\ X_i, else \end{cases} \quad (15)$$

Where, N_p denotes the number of recommended puzzle pieces, T denotes the iteration counter, T signifies the maximal amount of iterations, x_{i,d_j}^{new} represent the new value for d_j 'th dimension of i 'th puzzle, and x_{h,d_j} indicates the selected puzzle piece from h 'th puzzle that h is randomly chosen. Afterward upgrading each member of the population based on the initial and next phases, an iteration of the model, and the new status of the members of the population can be defined. Figure 2 illustrates the flowchart of POA.

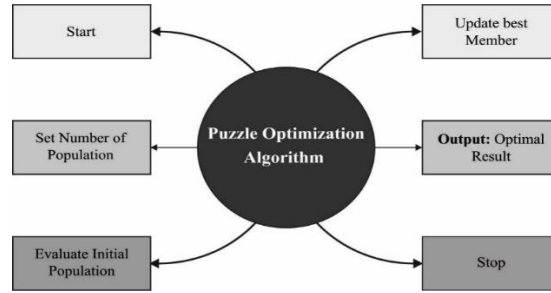


Figure 2. Flowchart of POA

2.4. Classification model

At the final stage, the C4.5 DT model is utilized for effective identification and recognition of Telugu characters. In the C4.5 DT approach experiments, the trained samples that are an identical outcome are eliminated as it is not most significant. Therefore, it is not be restricted in the DT once it does not have lesser 2 outcomes that are minimum count of samples. The candidate splits are occupied as for concerning during the case that it is cut a specific count of samples. There is an multi district litigation (MDL) based adjustment for separating numeric attributes. Quinlan planned heuristic to avoid over-fit. The next subtraction is to determine that information gain (IG) is negative. Once it doesn't have elements that are positive IG which is a variety of pre-pruning, the tree stops growing. It is signified as it can be unexpected to obtain a pruned tree but post-pruning is not active [22].

C4.5 employs the gain ratio:

$$P(D; D_1, \dots, D_k) = G(D; D_1, \dots, D_k) \cdot \left(- \sum_{i=1}^k \frac{|D_{ki}|}{D} \log \frac{|D_{ki}|}{D} \right)^{-1}, \tag{16}$$

which is distinct in the IG form, taking normalization on the count of feature values. Actually, the feature with maximal gain ratio, among features with higher-than-average IGs is selected as the separations.

3. EXPERIMENTAL VALIDATION

In this section, a detailed experimental validation of the POACN-TCR model is carried out using different aspects [23]. It is an integration of the 29,188 training instances gathered from 111 writers and 9,224 test instances gathered from 35 writers apart from the writers utilized for training data.

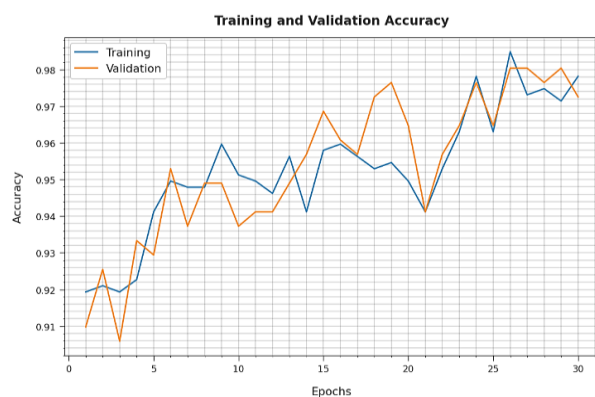


Figure 3. Accuracy graph analysis of POACN-TCR technique

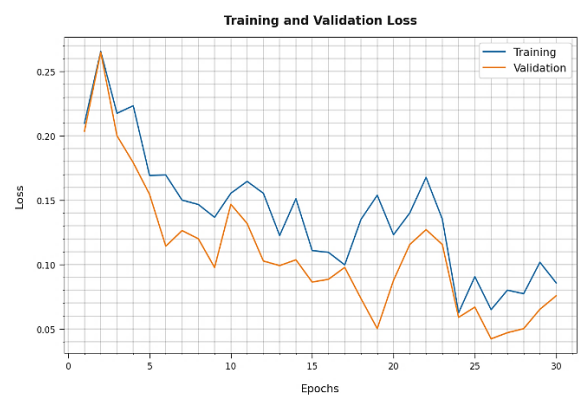


Figure 4. Loss graph analysis of POACN-TCR technique

Table 1 provides a detailed performance validation of the POACN-TCR model under distinct aspects. Figure 5 shows the sample size and recognized samples offered by the POACN-TCR model. The results indicated that the POACN-TCR model has offered increased number of recognized samples. On top-1 results, the POACN-TCR model has recognized 4,513 samples out of original 4,690 samples. Also, on top-2

results, the POACN-TCR method has recognized 2,927 samples out of original 3,083 samples. Besides, on top-3 results, the POACN-TCR technique has recognized 950 samples out of original 997 samples. In addition, on top-4 results, the POACN-TCR model has recognized 174 samples out of original 191 samples. At last, on top-5 results, the POACN-TCR system has recognized 18 samples out of original 19 samples.

Table 1. Performances of proposed POACN-TCR model in terms of recognition rate (%)

Methods	Samples size	Recognized samples	Recognition rate (%)
POACN-TCR 1	4690	4513	96.23
POACN-TCR 2	3083	2927	94.94
POACN-TCR 3	997	950	95.29
POACN-TCR 4	191	174	91.10
POACN-TCR 5	19	18	94.74
No. of samples	8980	8582	95.57

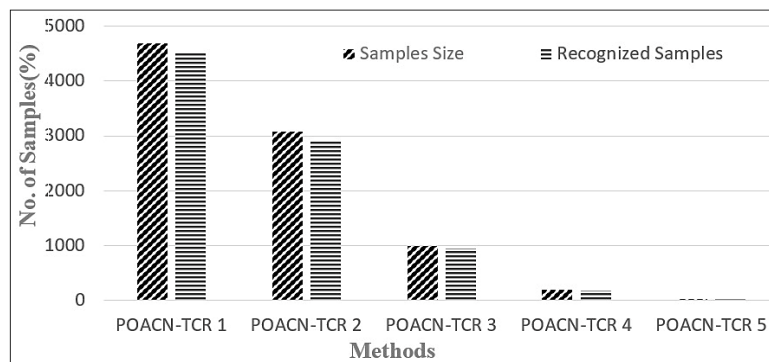


Figure 5. Result analysis of POACN-TCR technique

A detailed recognition rate examination of the POACN-TCR model under varying top-k results is given in Figure 6. The results indicated that the POACN-TCR model has showcased increased recognition rate. For instance, on top-1 results, the POACN-TCR model has offered recognition rate of 96.23%. At the same time, on top-3 results, the POACN-TCR algorithm has obtainable recognition rate of 95.29%. Meanwhile, on top-5 results, the POACN-TCR methodology has accessible recognition rate of 94.74%.

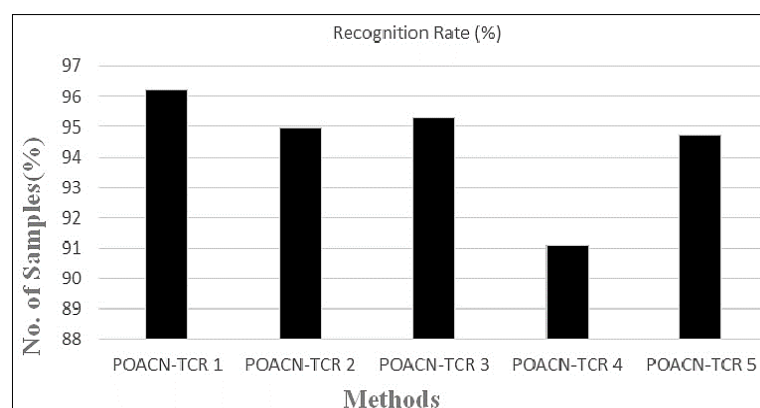


Figure 6. Recognition rate analysis of POACN-TCR technique

Finally, a comparative examination of the POACN-TCR model with recent models is provided in Table 2 and Figure 7 [24], [25]. The results indicated that the POACN-TCR model has demonstrated under distinct aspects. For instance, with top-1 results, the POACN-TCR model has offered increased recognition rate of 95.57% whereas the Babu et al. and Prasanth et al. models have provided reduced recognition rates of

91.60% and 90.60% respectively. Followed by, with top-3 results, the POACN-TCR approach has gained maximum recognition rate of 98.12% whereas the Babu et al. has accomplished minimal recognition rate of 97%. In addition, with top-5 results, the POACN-TCR approach has gained maximal recognition rate of 99.36% whereas Babu *et al.* [24] has accomplished reduced recognition rate of 98.70%. After examining the detailed results and discussion, it can be clear that the POACN-TCR model is found to be effective over the other methods.

Table 2. Results comparison of proposed POACN-TCR method with existing models

Performances	POACN-TCR	Babu <i>et al.</i>	Prasanth <i>et al.</i>
Top-1	95.57	91.60	90.60
Top-2	98.12	97.00	-
Top-3	98.78	98.00	-
Top-4	99.07	98.40	-
Top-5	99.36	98.70	-

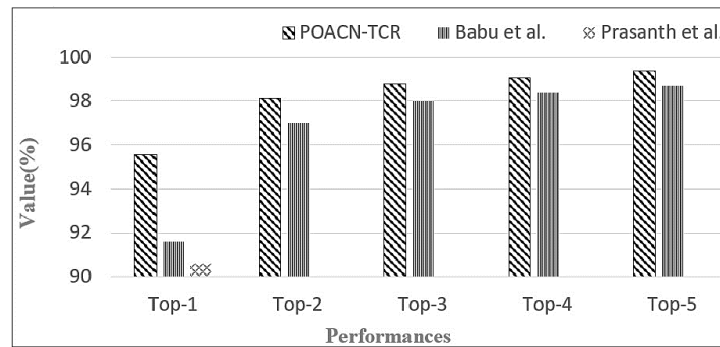


Figure 7. Comparative analysis of POACN-TCR technique with existing methods

4. CONCLUSION

In this study, a novel POACN-TCR model has been developed to effectively identify and recognize distinct Telugu characters in online. The POACN-TCR model primarily undergoes pre-processing in different ways such as normalization, smoothing, and interpolation. Followed by, the POACN-TCR model designs an effective CapsNet model to generate feature vectors and hyperparameter optimization take place using POA. Finally, the C4.5 decision tree classifier is utilized for the effectual recognition of Telugu characters. The utilization of POA for hyperparameter optimization of the CapsNet model helps in achieving improved recognition performance. For ensuring the improved outcomes of the POACN-TCR approach, a wide-ranging experimental analysis is performed and the results pointed out the betterment of the POACN-TCR model on the recent approaches. In future, hybrid DL techniques are employed for enhanced TCR performance.




REFERENCES

- [1] R. Aarthi, R. S. M. Varma, J. S. Vaishnavi, N. V. S. Prashanth, and G. T. Srikar, "Performance analysis of Telugu characters using deep learning networks," in *Lecture Notes in Electrical Engineering*, vol. 711, 2021, pp. 311–322.
- [2] N. Sarika and N. Sirisala, "Deep learning techniques for optical character recognition," in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 55, 2021, pp. 339–349, doi: 10.1007/978-981-15-8677-4_28.
- [3] B. S. Babu, S. Nalajala, K. Sarada, V. M. Naidu, N. Yamsani, and K. Saikumar, "Machine learning based online handwritten telugu letters recognition for different domains," in *Intelligent Systems Reference Library*, vol. 210, 2022, pp. 227–241.
- [4] N. Sarika, N. Sirisala, and M. S. Velpuru, "CNN based Optical character recognition and applications," in *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, Jan. 2021, pp. 666–672, doi: 10.1109/ICICT50816.2021.9358735.
- [5] K. R. Devarapalli and A. Negi, "Distributed training of deep neural network for segmentation-free Telugu word recognition," in *Machine Learning Technologies and Applications*, Springer, Singapore, 2021, pp. 113–120.
- [6] S. Inuganti, "Online handwritten Telugu character recognition using normalized differential chain code feature," *Webology*, vol. 18, no. SI04, pp. 01–15, Sep. 2021, doi: 10.14704/WEB/V18SI04/WEB18110.
- [7] S. J. Basha, D. Veeraiah, G. Pavani, S. T. Afreen, P. Rajesh, and M. S. Sasank, "A novel approach for optical character recognition (OCR) of handwritten telugu alphabets using convolutional neural networks," in *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Aug. 2021, pp. 1494–1500, doi: 10.1109/ICESC51422.2021.9532658.
- [8] S. K. Gorla, S. S. Tangeda, L. B. M. Neti, and A. Malapati, "Telugu named entity recognition using bert," *International Journal of Data Science and Analytics*, Jan. 2022, doi: 10.1007/s41060-021-00305-w.
- [9] A. T. Anju, B. P. Chacko, and K. P. M. Basheer, "Review of offline handwritten text recognition in south Indian languages," *Malaya Journal of Matematik*, vol. 9, no. 1, pp. 751–756, 2021, doi: 10.26637/MJM0901/0132.




- [10] S. Satla and S. Manchala, "Dialect Identification in telugu language speech utterance using modified features with deep neural network," *Traitement du Signal*, vol. 38, no. 6, pp. 1793–1799, Dec. 2021, doi: 10.18280/ts.380623.
- [11] S. S. Bhat, A. Ananth, R. Nambiar, and N. Bhat, "Building dataset and deep learning-based inception model for the character classification of Tigarariscrypt," pp. 239–252, 2022, doi: 10.1007/978-981-16-3342-3_20.
- [12] C. Prathima and N. B. Muppalaneni, "Deep learning approach for prediction of handwritten Telugu vowels," in *Smart Innovation, Systems and Technologies*, vol. 213 SIST, 2021, pp. 367–374.
- [13] A. Jaisswal, P. Ghalsasi, A. Jain, M. Batra, and S. Arora, "Classification of handwritten indic scripts using deep learning," in *Lecture Notes in Networks and Systems*, vol. 333, 2022, pp. 499–506.
- [14] R. Guha, N. Das, M. Kundu, M. Nasipuri, and K. C. Santosh, "DevNet: an efficient CNN architecture for handwritten devanagari character recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 34, no. 12, p. 2052009, Nov. 2020, doi: 10.1142/S0218001420520096.
- [15] A. K. Pujari, C. D. Naidu, M. S. Rao, and B. C. Jinaga, "An intelligent character recognizer for Telugu scripts using multiresolution analysis and associative memory," *Image and Vision Computing*, vol. 22, no. 14, pp. 1221–1227, Dec. 2004, doi: 10.1016/j.imavis.2004.03.027.
- [16] P. Sujatha and D. L. Bhaskari, "Telugu and Hindi script recognition using deep learning techniques," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 11, pp. 1758–1764, Sep. 2019, doi: 10.35940/ijitee.K1755.0981119.
- [17] A. Angadi, V. Kumari, and S. Keerthi, "A deep learning approach to recognize handwritten Telugu character using convolution neural networks," *International Journal of Information Systems & Management Science*, vol. 1, no. 2, pp. 8–12, 2018.
- [18] S. T. Soman, A. Nandigam, and V. S. Chakravarthy, "An efficient multiclassifier system based on convolutional neural network for offline handwritten Telugu character recognition," in *2013 National Conference on Communications (NCC)*, Feb. 2013, pp. 1–5, doi: 10.1109/NCC.2013.6488008.
- [19] M. Jangid and S. Srivastava, "Handwritten Devanagari character recognition using layer-wise training of deep convolutional neural networks and adaptive gradient methods," *Journal of Imaging*, vol. 4, no. 2, p. 41, 2018, doi: 10.3390/jimaging4020041.
- [20] R. Mukhometzianov and J. Carrillo, "CapsNet comparative performance evaluation for image classification," *arXiv preprint*, May 2018, [Online]. Available: <http://arxiv.org/abs/1805.11195>.
- [21] F. A. Zeidabadi and M. Dehghani, "POA: Puzzle optimization algorithm," *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 1, Feb. 2022, doi: 10.22266/ijies2022.0228.25.
- [22] R. Benkercha and S. Moulahoum, "Fault detection and diagnosis based on C4.5 decision tree algorithm for grid connected PV system," *Solar Energy*, vol. 173, pp. 610–634, Oct. 2018, doi: 10.1016/j.solener.2018.07.089.
- [23] Lipi, "Telugu Character Dataset," *Lipi ToolKit*, 2013. <http://lipitk.sourceforge.net/datasets/teluguchardata.htm>.
- [24] V. Babu, L. Prasanth, R. Sharma, G. V. Rao, and A. Bharath, "HMM-Based online handwriting recognition system for Telugu symbols," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, Sep. 2007, vol. 1, pp. 63–67, doi: 10.1109/ICDAR.2007.4378676.
- [25] L. Prasanth, V. Babu, R. Sharma, G. V. Rao, and D. M., "Elastic matching of online handwritten tamil and telugu scripts using local features," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, Vol 2, Sep. 2007, vol. 2, pp. 1028–1032, doi: 10.1109/ICDAR.2007.4377071.

BIOGRAPHIES OF AUTHORS



Josyula Siva Phaniram    has completed Master of Science in Computer Science from Nagarjuna University and Master of Technology in Information Technology from Karnataka state university. He is currently pursuing Ph.D. in Computer Science at Krishna University, Machilipatnam. His research interest includes artificial intelligence, machine learning, deep learning. He can be contacted at email: phaniram.research@gmail.com.



Dr. Mukkamalla Babu Reddy    has completed his Post Graduation and Doctoral degree from Acharya Nagarjuna University, AP. He is activity involved in teaching and research for more than two decades. He has published more than 60 research papers in reputed journals and guiding PhD Scholars in Computer Science & Engineering domain. At present, he is working as Principal, University College of Engineering and Technology, Krishna University, Machililpatnam, AP, India. He can be contacted at email: m_babureddy@yahoo.com.