Journal of Positive School Psychology
2022, Vol. 6, No. 5, 5307–5318 http://journalppw.com

Design of Optimal Deep Learning Assisted Online Telugu Character Recognition Model

Josyula Siva Phaniram[1], Dr. M. Babu Reddy[2]

[1] Research Scholar,Department of Computer Science,Krishna University, Machilipatnam.
[2] Principal, FAC, Krishna University, Machilipatnam
[1] phaniram.research@gmail.com, [2] mbreddy.cs@kru.ac.in

Abstract

Telugu character recognition process has received significant attention due to the exponential utilization of resources like images, smartphones, iPods, and paper documents. It can be divided into two types namely offline character recognition and online character recognition. Offline character recognition is a process of identifying Telugu characters from the scanned image or document whereas online character recognition enables to recognition of characters by the machine while the user writes. Several researchers have attempted to design online Telugu character recognition model by the use of distinct classification models and feature extraction approaches; however, the performance is yet to be improved. In this aspect, this study focuses on the design of optimal deep learning based online Telugu character recognition (ODL-OTCR) model. The goal of the ODL-OTCR technique is to recognize as well as classify the Telugu characters in online model. The ODL-OTCR technique involves data preprocessing to preprocess the character stroke in three ways namely normalization, smoothing, and interpolation. Besides, a beetle swarm optimization (BSO) with EfficientNet model is utilized as a feature extractor and finally, Siamese Neural Networks (SNN) model is employed for the classification process. In order to showcase the improved performance of the ODL-OTCR technique, a series of simulations take place and the results are inspected interms of different aspects. The simulation results highlighted the betterment of the proposed ODL-TCR technique over the recent techniques.

Keywords— Telugu character recognition, Deep learning, Parameter tuning, Machine learning, Preprocessing, EfficientNet

1. Introduction

In modern times, there is a growing demand for applications which could identify hand-written characters automatically. At present, there is a huge demand to store this type of data available in books or in any kind of document to a computer for upcoming work that is later retrieved by a search method [1]. It could be achieved by scanning the data that has to save and store. However, the problems can be faced with complexity for reading the documents as we could not search the content of an image. This is due to the fact that hand-written font character differs from the fonts of character in computer systems. Therefore, the computer failed to identify the character when reading them. Optical Character Recognition (OCR) is employed for classifying optical patterns to respective alphanumerical characters/another character. This is a kind of document image analysis in which scanned digital images have a handwritten script/machine printed documents are inputted to an OCR software engine and translate to modifiable machine readable digital text formats (such as ASCII texts). The OCR is a leading approach in computer vision [2]. This could simply attain data from different images, however, it could not identify Telugu handwritten characters because of an absence of dataset and trained deep convolution neural network. OCR is the procedure of translating scanned images of printed, hand-written, typed text to machine encoded texts. OCR technique addresses the issue of identifying each kind of distinct character [3]. Printed and Handwritten characters could be identified and transformed to a digital data/machine readable format. Through OCR, we can convert sequential code/numbers consist of letters and numbers to a digital output.

Telugu language is spoken mainly in the state of Telangana and Andhra Pradesh India. It has the 3rd larger amounts of native speakers in India [4]. They are written in scripts that originate from the Brahmi script, Telugu is a south central Dravidian language based on Urdu, Prakrit, and Sanskrit. The symbol in the Telugu languages is divided broadly into the subsequent classes like Conjunct consonants, Vowels, Consonants, and Vowel diacritics. An individual Telugu character (Aksharam) in Indian languages generally represents whole syllables [5]. An individual Telugu character consists of i) more than one consonants ii) pure vowel. Due to this fact, the overall amount of consonants and vowels were counted to be fifty two. The complication in character identification differs between various languages because of its different strokes, shapes, and numbers of characters.

Telugu Character Recognition is separated into 2 kinds 1) Online Handwritten Character Recognition 2) Offline Handwritten Character Recognition. An online hand-written character is an approach of identifying characters through a machine when the user writes [6]. The Offline Hand-written character recognition is a process of recognizing the scanned hand-written images/documents; now handheld device is employed to identify the characters by recording (x, y) coordinate of the track of characters. Telugu character image is recognized by any touch screen devices or some special digitizer or PDA, sensors pick up the pen tip movement and pen up or pen down switching's and users written stroke is considered as a sampling capture the pens (x, y) coordinate at equally spaced time interval pen down and pen up of recognizing Telugu character image. Online Telugu character identification has become an active and challenging area in the research of image processing and pattern recognition because of the increasing usage of the assets like photographs, smart phones, iPad, and paper documents.

The most important challenges in online hand-written character identification for Indian language is to construct a scheme which is capable of distinguishing among variations in writing the similar strokes (while the similar strokes are written by a distinct writer or the similar writer at distinct times) and minor variations in same character in the scripts [7]. Other problems met could be attributed to the huge amount of stroke and character classes. Many Indian scripts consist of characters modifier arising in many nonoverlapping horizontal units that are located on one or both sides of the primary consonant. In this case, they should monitor the series of horizontal units as they are written.

This study proposes an optimal deep learning based online Telugu character recognition (ODL-OTCR) model to recognize as well as classify the Telugu characters in online model. The ODL-OTCR technique involves data preprocessing to preprocess the character stroke in three ways namely normalization, smoothing, and interpolation. Moreover, a beetle swarm optimization (BSO) with EfficientNet model is utilized as a feature extractor. Furthermore, Siamese Neural Networks (SNN) model is employed for the classification process. For examining the effectual outcomes of the ODL-OTCR technique, a comprehensive experimental validation process is carried out and the results are examined under various dimensions.

2. Literature Review

In Aarthi et al. [8], a CNN method is presented to recognize fifty two Telugu characters. The performance of the presented method was calculated using the optimizers like SGD, Adam, Adagrad, and Adadelta. Also, the preprocessing steps are assisted in enhancing the performance. The CNN models are related to the VGG-16 and obtained lesser accuracy. Muppalaneni [9] deploys DL method for enhancing the performance i.e., examined training accuracy and test accuracy as 96.13% and 79.61% correspondingly. They constructed an ML model using CNN method for Telugu Hand-written Gunithalu. They generated specific datasets and it is accessible in IEEE Dataports.

Chauhan et al. [10] proposed a novel deep learning architecture which exploits transfer learning and image-augmentation for end-to-end learning for script independent handwritten character recognition, called HCR-Net. The network is based on a novel transfer learning approach for HCR, where some of lower layers of a pre-trained VGG16 network are utilized. Due to transfer learning and image-augmentation, HCR-Net provides faster training, better performance, and better generalizations. Ramana Murthy et al. [11] employed OCR Scheme and different ML techniques that as CNN and SVM. In SVM, they created a stroke detection engine and the character has been denoted as a series of strokes, and features were classified and extracted. The SVM model for Telugu language (south Indian) character detection method using classification of time and higher detection accuracy and minimal training. A qualified analysis has been executed for testing the efficacy of the presented model against prior method.

Cheekati and Rajeti [12] deal with emerging a reliable, fast Telugu hand-written ResNet for offline and online character detection as well as improve the classification accuracy. The presented method depends on a systematic approach for recognizing online and offline Telugu hand-written characters through residual learning frameworks named ResNet. An RL network is a model of DNN in which the training of data is highly efficient. ResNet allows making DNN model by tackling the gradient reducing problems which occur in DCNN model. Guha et al. [13] distinct CNN methods on publicly accessible hand-written Devanagari character and numeral dataset. It is mainly concentrating on relative works by considering training time, memory consumption, and trainable parameters. They, design and propose DevNet, an adapted CNN framework which produces possible results, because memory space and computation complexity are the major concern in this model.

3. The Proposed ODL-OTCR Technique

In this study, a new ODL-OTCR technique is designed to recognize and classify Telugu characters in online model. The ODL-OTCR technique involves different stages of operations namely preprocessing, EfficientNet based feature extraction, BSO based hyperparameter optimization, and SNN based classification. Fig. 1 illustrates the overall process of ODL-OTCR model. The detailed working process involved in the ODL-OTCR technique is elaborated in the following sections.

3.1. Preprocessing

The input to the identification scheme includes features of the stroke in all the written characters. The SNN based stroke identification model was taken into account due to its generalization ability on huge dimensional data. The identified stroke, gathered according to the proximity analyses, are mapped to character through data on effective strokes combination for the scripts. The output of detection engines is a series of identified characters. All the characters are denoted as a combination of strokes. A stroke is determined as the trajectory traced by the pen from a pen down events to a pen up events as well it is denoted by the information taken as the strokes are written. The number of gathered points differs by the speed and stroke of writing. The representation of the strokes is selected as a fixed/variable length. Fixed length formats require the feature vector representing the strokes as a constant length. Because the data must contain a constant length representation in the SNN based method, fixed length feature vector was employed for representing the stroke. The data of strokes can be pre-processed in 3 steps are described below.

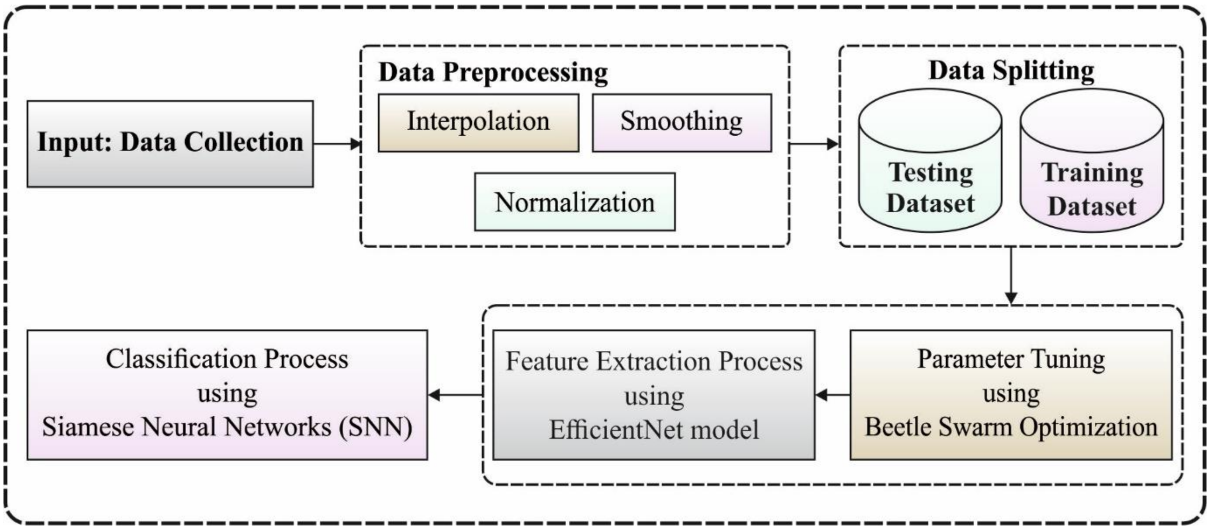

Fig. 1. Overall block diagram of ODL-OTCR model

Normalization

Few characters could be written by several strokes with some strokes expanding below/above the primary part of the character. The primary part of a character provides a measure of the character size and line space utilized by the writer. In general, there would be at least one stroke in the primary portion of the character where this data is attained. In case where it is impossible to recognize the primary stroke, the height of the characters is employed to the normalization. After normalization, the strokes are uniformly sampled alongside the length of its curve.

Smoothing

The normalized strokes are yet piece-wise linear, through line segment linking the point with its curve. Gaussian filters are applied for smoothing the stroke. A 1D Gaussian distribution is taken into account by the smoothing kernels that are initially convolved with the x-axis as well as y-axis. The smoothing is attained based on the size of window function and the smoothing kernel parameter employed.

Interpolation

The last step in pre-processing is where the smoothened strokes are interpolated for providing a fixed number of points, equally spaced with the curve length. The numbers of points are selected according to the average number of points for each stroke in the provided datasets.

3.2. Optimal EfficientNet based Feature Extraction

The preprocessed data is employed in the EfficientNet model to extract a useful set of feature vectors. EfficientNets are a list of classifiers introduced recently in 2019 and depends on Compound Scaling and AutoML. The compound scaling methods are applied for scaling up this baseline to attain EfficientNet-B1 to B7. Subsequently, AutoML is employed for developing mobile size baseline network (EfficientNet-B0). The Compound Scaling methods uniformly scale each dimension of resolution, depth, and width through simple but very efficient compound coefficients. The width of layer must rise 10%, the image resolution 15%, and the depth 20%, for keeping all things as effective as possible, when improving the accuracy and increasing the performance. Gamma, Alpha, and beta are the scaling multiplier for resolution, depth, and width, correspondingly. They are attained by a grid search. Phi is a user specific coefficient. It is a real number that controls resources. Below are the equations of depth, weight, and resolution based on Phi:

$$Depth: d = \alpha^{\phi}, \quad (1)$$

$$Width: w = \beta^{\phi}, \quad (2)$$

$$Resolution: r = \gamma^{\phi}, \quad (3)$$

$$while: \alpha.\beta^2.\gamma^2 \approx 2;\ \alpha \geq 1, \beta \geq 1\ and\ \gamma 1. \quad (4)$$

The efficientNet-B0 structure has mobile sized framework containing 11M trainable parameters. In all rows are separating stages i in the network [14]. All stages i is characterized by the amount of layers $\hat{L}_i$, an input solution size $< \hat{H}_i, \hat{W}_i >$ and output channels size $\hat{C}_i$.

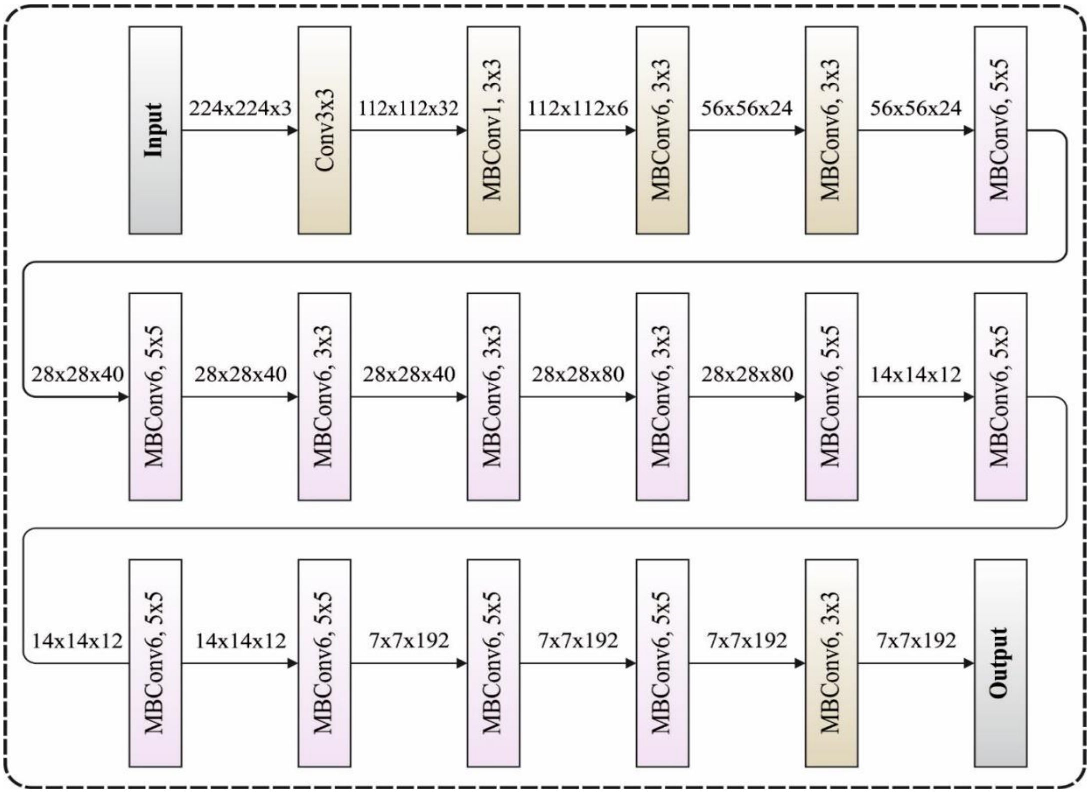

Fig. 2. Architecture of EfficientNet

It uses seven inverted residual blocks. Squeeze and excitation blocks are used along with the swish activation function. EfficientNet uses 7 MBConv blocks. Every MBConv block takes 2 inputs. The initial one is data, and the next one is block argument. The data is received from the final layers. Fig. 2 demonstrates the framework of EfficientNet. The block arguments are a group of features to be employed inside MBConv blocks, such as output filter, input filter, squeeze ratio, expansion ratio, and so on. The expansion phase aims to expand the layer to make it wide. The depth-wise convolution phase applies a depth-wise convolution using the kernel size mentioned in the block arguments. The excitation and Squeeze phase extracts the global feature using the global average pooling. Then, it squeezes the number of channels using the squeeze ratio. The Output phase applies convolution operation using the output filters mentioned in the block arguments.

For optimally adjusting the hyperparameters involved in the EfficientNet model, the BSO algorithm is employed. The BSO approach was assumed that an optimization model which combines the method of beetle foraging as well as PSO manner. Many researches are projected that two 2 antennae of beetles are utilized for determining adjacent regions. But, an antenna detects extremely concentrated food aromas on one side, afterward, a beetle will travel nearby the antennae at existing side. According to easy biological nature, a meta-heuristic optimized manner dependent upon foraging character of beetles is resultant. It can be explained with the place of beetle from S-dimension space at t as x^t, and place position of beetle at t+1 is maintained in (5).

$$x^{t+1} = x^t + \delta^t * b * sign\left(f(x_{rt}) - f(x_{lt})\right) \quad (5)$$

$$x_{rt} = x^t + d^t * \vec{b}; x_{lt} = x^t - d^t * \vec{b} \quad (6)$$

$\vec{b} = \frac{rands(s,1)}{\|rands(s,1)\|}$; $\delta^t = 0.95\delta^{t-1}$;

$$d^t = 0.95 d^{t-1} + 0.01 \tag{7}$$

At this point, $\vec{b}$ refers the random explore way of beetle in S-dimension space, and *rands* $(.)$ represents the arbitrary expression. δ^t signifies the exploring length of beetle, but d^t denotes the noticeable distance of antennae. A few initial values of δ^t and d^t are frequently set as maximal and decreased gradually. An important objective of this creation is for developing past searching restrict in optimized procedures for taking on enormous regions and locate local extremes. x_{lt} and x_{rt} are termed as forecasted places of left as well as right antennae of beetles, and high intensity of food flavor at 2 places are projected as f(x_{rt}) and f(xlt), in which the fitness function (FF) rates of projected technique sign $(.)$ refers the symbol expression.

The BSO technique was deficient bur operate high-definition tasks, and performance of an iteration is maximal vulnerability to initial place of antennae. Similar to PSO technique, all beetles existing in BSO manner represents the able solutions of optimized problem, and many beetles allocate data particularly. The procedure of enhancing the speed of beetle is measured with the trend of all beetles seeking to single extreme and tendency of BSO observing to global extreme rate. Different, the place update of all beetles could not be estimated with increasing the speed upgrading and information attained with equivalent antennae. Therefore, it executes loosening approach to smarter computing methods. $\mathrm{X} = (X_1, X2, \ldots, X_n)$ was utilized for representing the beetle swarm together with population size of n in S-dimension search space, and $X_i = (x_{i1}, x_{i2}, \cdots, x_{iS})^T$ demonstrates the S-dimension vector and place parameter of beetle i from S-dimensional searching space and able solution of optimized challenge. $V_i = (v_{i1}, v_{i2}, \ldots, v_{iS})^T$ implies the speed variable of beetle i. The single extreme is projected by $P_i = (p_{i1}, p_{i2}, \cdots, p_{iS})^T$, as well as global extreme was understood by $P_g = \left(p_{g1}, p_{g2}, \cdots, p_{gS}\right)^T$. So, the upgrades of speed and place of BSO structure is given as:

$$x_{is}^{k+1} = x_{is}^{k} + \lambda v_{is}^{k} + (1 - \lambda)\xi_{is}^{k}, \tag{8}$$

$$v_{is}^{k+1} = \omega v_{is}^{k} + c_1 r_1 \left(p_{is}^{k} - x_{is}^{k}\right) + c_2 r_2 \left(p_{gs}^{k} - x_{gs}^{k}\right), \tag{9}$$

$$\xi_{is}^{k+1} = \delta^k * v_{is}^{k} * sign\left(f\left(x_{rs}^{k}\right) - f\left(x_{ls}^{k}\right)\right), \tag{10}$$

$$x_{rs}^{k+1} = x_{rs}^{k} + v_{is}^{k} * \frac{\vec{d}}{2};\ x_{ls}^{k+1} = x_{ls}^{k} - v_{is}^{k} * \frac{\vec{d}}{2}, \tag{11}$$

where $s = 1, 2, \cdots, S; i = 1, 2, \ldots, n;$ k refers the processing time. ξ_{is} signifies the single portion of displacement calculated with robust data defined as beetle antennae that are regarded as alternate portion of displacement increment [15]. Afterward, the loosening factor (λ) in (8) and inertia weight (ω) in Eq. (9) is adjusted, therefore. r_1 and r_2 are named as random functions with the value range in [0, 1]. The attribute c_1 and c_2 evaluate the result of single as well as global extremes of beetles. The semantics of δ^k, d, x, and $f(x)$ are same as with basics of BAS. During BSO method, the place upgrade works utilizing exploring procedure of beetle monomer and recognizes the extended rule of PSO approach.

3.3. SNN based Classification

Finally, the extracted feature vectors are fed into the SNN model to recognize and classify the Telugu characters. The SNN is a feedforward network with error BP. The network composes of 2 indistinguishable FFNN combined its outcome. Amongst training, all the networks examine a profile comprised of genuine values and procedures their values at all layers. The network performs some of the neuron dependent upon this value as well as upgrade their weights with error BP, and at the finish, it makes a resultant profile which has related to the output of another network. The SNN relates the outcome of upper as well as lower networks with computing the distance metric. With this distance, the network conditions that 2 outputs were distinctive/comparative. This technique again names happening as positive/negative, according to the distance metric. The last outcome is at last be related to their equivalent ground truth values.

The projected SNN structure has twin NN (top as well as bottom networks), in which each one contains 3 dense layers (another than the input layer). An input layer contains 64 neurons, and all other layer contains 128 neurons [16]. The activation function executes to initial 3 layers is "$tanh$". For avoiding over fit, the dropout was executed amongst layer1 and layer2, and layer2 and layer3 with fraction of 10%. The 2 twin NN outputs are combined to procedure the resultant layer with 1 neuron, in which the Euclidean distance has been implemented amongst the outputs of top as well as bottom networks for measuring the similarity amongst the 2 outputs. The dataset was divided into 2 partitions so that two-thirds of dataset are offered to train and the rest is given to test. The data is fitting to the technique from batches. The network has been training with 200 epochs. An objective of network was to minimize positive pairs distance but maximized negative pairs distance. These are completed utilizing the contrastive loss functions:

$$L = \begin{cases} D(X, Y_p) & \text{for positive pairs,} \\ \max(0, m - D(X, Y_n & \text{for negative pairs,} \end{cases} \quad (12)$$

where L refers the loss functions, (X, Y_p) signifies the positive pair, (X, Y_n) represents the negative pair, D implies the distance amongst 2 records of similar pair, and m denotes the margin value that illustrates that 2 records of negative pair were distinctive sufficient. For getting the values of network weight at that loss was optimal, the RMSprop optimized technique was executed.

4. Performance Validation

This section investigates the performance of the ODL-TCR technique on the applied HP Labs dataset [17]. It is an integration of the 29188 training instances gathered from 111 writers and 9224 test instances gathered from 35 writers apart from the writers utilized for training data. The symbols are split into 5 distinct groups depending upon the number of constituent strokes. With the consideration of every probable way of writing each of the characters belongs to 141 class labels, an interval is represented by the inclusion of the predefined number of minimum and maximum constituent strokes. It has a total of 28482 and 8980 samples under training and testing set.

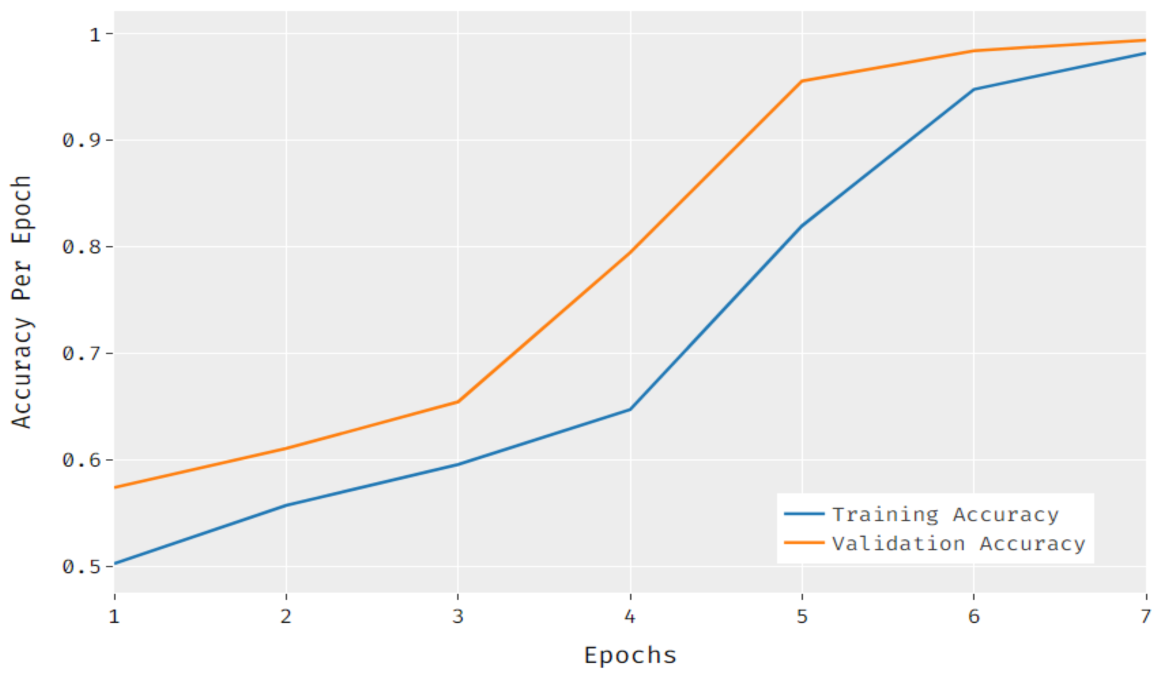

Fig. 3. Accuracy analysis of ODL-TCR model

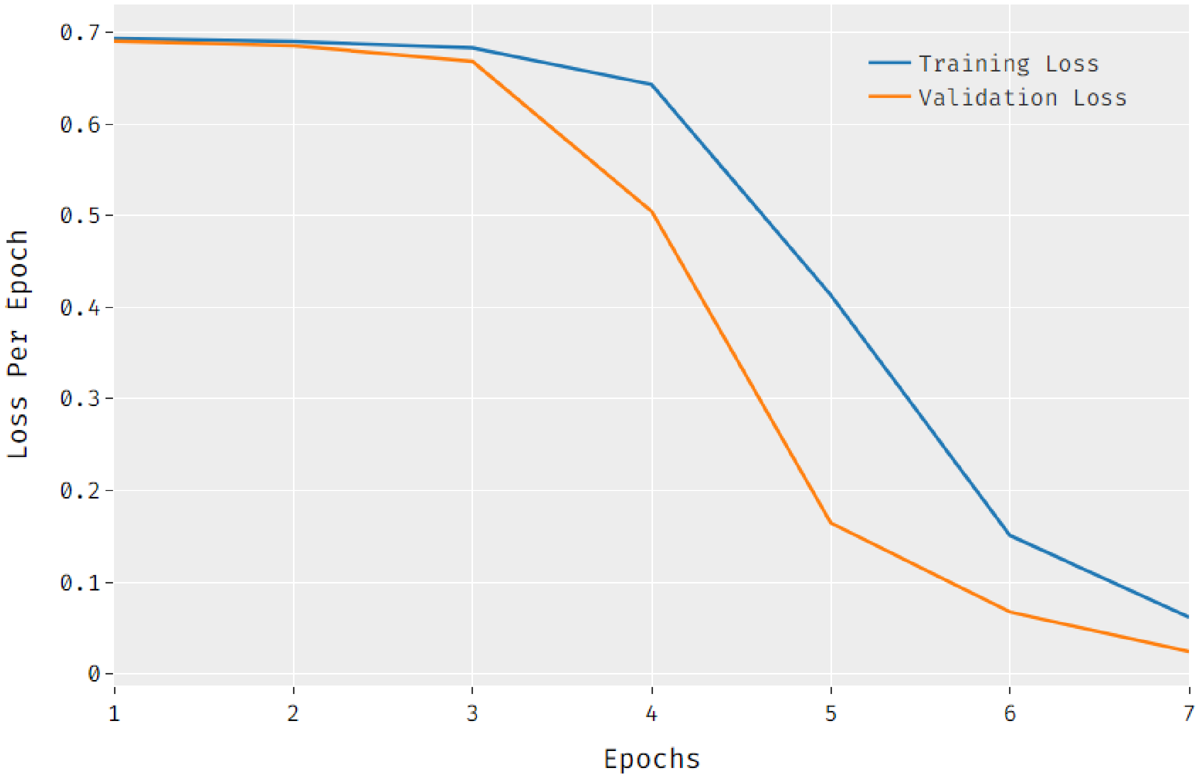

Fig. 4. Loss analysis of ODL-TCR model

Fig. 3 illustrates the accuracy analysis of the ODL-TCR technique with existing techniques. The figure has shown that the ODL-TCR technique has accomplished higher training and validation accuracy. In addition, it is noted that the ODL-TCR technique has attained an increased validation accuracy compared to training accuracy.

Fig. 4 showcases the loss analysis of the ODL-TCR approach with recent algorithms. The figure demonstrated that the ODL-TCR manner has accomplished lower training and validation loss. Also, it can be clear that the ODL-TCR manner has obtained a minimal validation loss associated with training loss.

Table 1 Performances of Proposed ODL-OTCR Model in terms of Recognition Rate (%)

Methods	**Samples Size**	**Recognized Samples**	**Recognition Rate (%)**
ODL-OTCR 1	4690	4470	95.31
ODL-OTCR 2	3083	2878	93.35
ODL-OTCR 3	997	910	91.27
ODL-OTCR 4	191	168	87.96
ODL-OTCR 5	19	18	94.74
No. of Samples	**8980**	**8444**	**94.03**

Table 1 offers the results analysis of the ODL-TCR technique with existing technique interms of different measures. Fig. 5 investigates the ODL-TCR technique with existing techniques interms of recognized samples. The figure showcased that the ODL-TCR technique has accomplished maximum recognized samples.

For instance, the ODL-TCR-1 technique has showcased effective outcomes with the higher recognized samples of 4470. Besides, the ODL-TCR-2 manner has outperformed effectual results with the superior recognized samples of 2878. Along with that, the ODL-TCR-3 method has demonstrated efficient outcomes with the higher recognized samples of 910. Moreover, the ODL-TCR-4 methodology has depicted effective outcomes with increasingly recognized samples of 168. Furthermore, the ODL-TCR-5 methodology has exhibited performance results in the maximum recognized samples of 18.

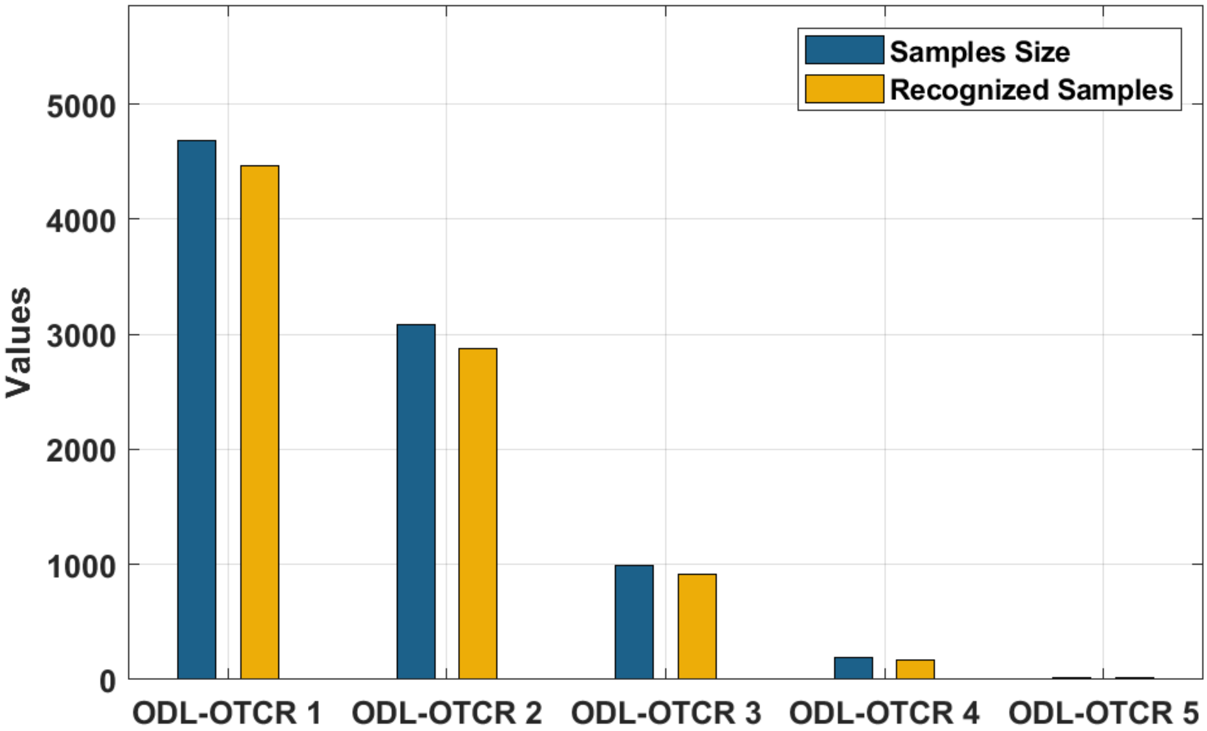

Fig. 5. Result analysis of ODL-TCR model

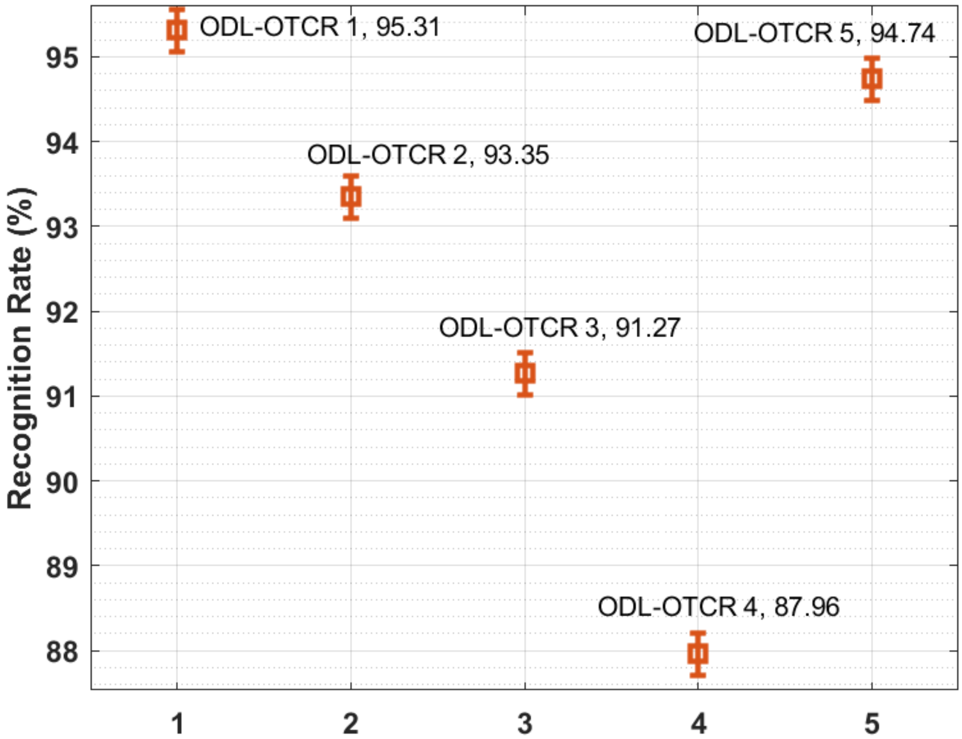

Fig. 6. Recognition rate analysis of ODL-TCR model

Fig. 6 examines the ODL-TCR manner with recent algorithms with respect to recognition rate. The figure portrayed that the ODL-TCR manner has accomplished maximal recognition rate. For instance, the ODL-TCR-1 algorithm has demonstrated effective outcome with superior recognition rate of 95.31%. Likewise, the ODL-TCR-2 method has showcased effectual outcomes with a maximal recognition rate of 93.35%. Similarly, the ODL-TCR-3 technique has outperformed effective outcomes with an increased recognition rate of 91.27%. Moreover, the ODL-TCR-4 methodology has showcased effective outcomes with a higher recognition rate of 87.96%. Furthermore, the ODL-TCR-5 manner has exhibited efficient outcomes with a superior recognition rate of 94.74%.

Table 2 Results comparison of proposed ODL-OTCR method with existing models

Performances	**ODL-OTCR**	**Babu et al.**	**Prasanth et al.**
Top-1	0.9403	0.9160	0.9060
Top-2	0.9725	0.9700	-
Top-3	0.9832	0.9800	-
Top-4	0.9894	0.9840	-
Top-5	0.9912	0.9870	-

In order to showcase the improved outcome of the ODL-TCR technique, a brief comparison study is made interms of recognition rate as shown in Table 2 and Fig. 7 [18, 19]. The ODL-TCR technique has outperformed the other techniques under top-5 executions. For instance, under top-1, the ODL-TCR technique has offered an increased detection rate of 0.9403 whereas the Babu et al. and Prasanth et al. techniques have obtained a reduced detection rate of 0.9403 and 0.9160. Meanwhile, under top-2, the ODL-TCR manner has existed a maximum detection rate of 0.9725 whereas the Babu et al. approach has reached a minimal detection rate of 0.9700. Eventually, under top-3, the ODL-TCR algorithm has accessible an increased detection rate of 0.9832 whereas the Babu et al. manner has reached a reduced detection rate of 0.9800. In line with, under top-4, the ODL-TCR methodology has obtainable an increased detection rate of 0.9894 whereas the Babu et al. technique has achieved a lower detection rate of 0.9840. Lastly, under top-5, the ODL-TCR approach has existing an enhanced detection rate of 0.9912 whereas the Babu et al. technique has gained a decreased detection rate of 0.9870.

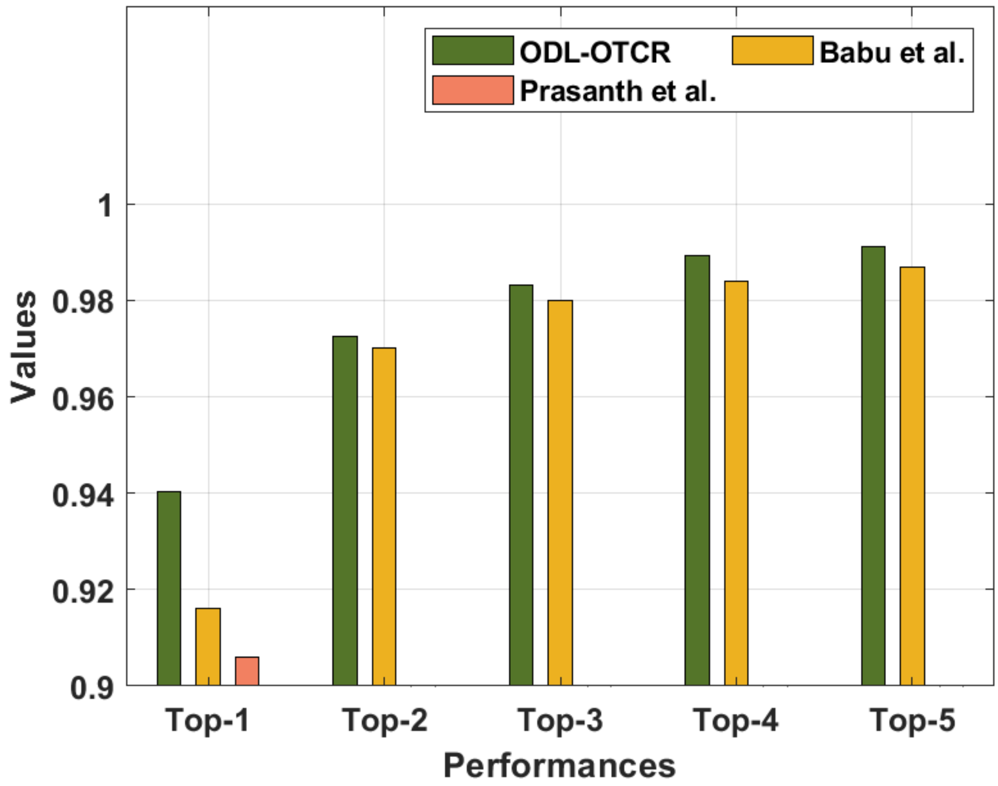

Fig. 7. Comparative analysis of ODL-TCR model

5. Conclusion

In this study, a new ODL-OTCR technique is designed to recognize and classify Telugu characters in online model. The ODL-OTCR technique involves different stages of operations namely preprocessing, EfficientNet based feature extraction, BSO based hyperparameter optimization, and SNN based classification. The application of BSO algorithm to fine tune the selection of hyperparameters involved in the EfficientNet model helps to considerably improve the overall classification outcome. For examining the effectual outcomes of the ODL-OTCR technique, a comprehensive experimental validation process is carried out and the results are examined under various dimensions. The simulation results highlighted the betterment of the proposed ODL-TCR technique over the recent techniques. In future, the performance of the ODL-OTCR technique can be increased by the use of DL based classification model and also deploy in a smartphone environment.

References

1. Swethalakshmi, H., Jayaraman, A., Chakravarthy, V.S. and Sekhar, C.C., 2006, October. Online handwritten character recognition of Devanagari and Telugu Characters using support vector machines. In *Tenth international workshop on Frontiers in handwriting recognition*. Suvisoft.
2. Soman, S.T., Nandigam, A. and Chakravarthy, V.S., 2013, February. An efficient multiclassifier system based on convolutional neural network for offline handwritten Telugu character recognition. In *2013 National Conference on Communications (NCC)* (pp. 1-5). IEEE.
3. Rajkumar, J., Mariraja, K., Kanakapriya, K., Nishanthini, S. and Chakravarthy, V.S., 2012, September. Two schemas for online character recognition of Telugu script based on support vector machines. In *2012 International Conference on Frontiers in Handwriting Recognition* (pp. 565-570). IEEE.
4. Sastry, P.N., Lakshmi, T.V., Rao, N.K., Rajinikanth, T.V. and Wahab, A., 2014, October. Telugu handwritten character

recognition using zoning features. In *2014 International Conference on IT Convergence and Security (ICITCS)* (pp. 1-4). IEEE.

5. Prakash, K.C., Srikar, Y.M., Trishal, G., Mandal, S. and Channappayya, S.S., 2018, October. Optical character recognition (ocr) for telugu: Database, algorithm and application. In *2018 25th IEEE International Conference on Image Processing (ICIP)* (pp. 3963-3967). IEEE.
6. Kumar, K.V. and Rao, R.R., 2013. Online handwritten character recognition for Telugu language using support vector machines. *International Journal of Engineering and Advanced Technology*, *3*(2), pp.189-192.
7. Jyothi, J., Manjusha, K., Kumar, M.A. and Soman, K.P., 2015. Innovative feature sets for machine learning based Telugu character recognition. *Indian Journal of Science and Technology*, *8*(24), p.1.
8. Aarthi, R., Varma, R.S., Vaishnavi, J.S., Prashanth, N.V.S. and Srikar, G.T., 2021. Performance analysis of telugu characters using deep learning networks. In Advances in Electrical and Computer Technologies (pp. 311-322). Springer, Singapore.
9. Muppalaneni, N.B., 2020, February. Handwritten Telugu compound character prediction using convolutional neural network. In 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE) (pp. 1-4). IEEE.
10. Chauhan, V.K., Singh, S. and Sharma, A., 2021. HCR-Net: A deep learning based script independent handwritten character recognition network. arXiv preprint arXiv:2108.06663.
11. P.V. Ramana Murthy, P.Andrews Hima Kiran, S. Ajay Kumar, Pattola Srinivas, ONLINE TELUGU HANDWRITTEN CHARACTER RECOGNITION USING EFFICIENT MACHINE LEARNING APPROACHES, International Journal of Future Generation Communication and Networking, Vol. 13 No. 4 (2020)
12. Cheekati, B.M. and Rajeti, R.S., 2020, October. Telugu handwritten character recognition using deep residual learning. In 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) (pp. 788-796). IEEE.
13. Guha, R., Das, N., Kundu, M., Nasipuri, M. and Santosh, K.C., 2020. Devnet: an efficient cnn architecture for handwritten devanagari character recognition. International Journal of Pattern Recognition and Artificial Intelligence, 34(12), p.2052009.
14. Ben Jabra, M., Koubaa, A., Benjdira, B., Ammar, A. and Hamam, H., 2021. COVID-19 Diagnosis in Chest X-rays Using Deep Learning and Majority Voting. *Applied Sciences*, *11*(6), p.2884.
15. Singh, P., Kaur, A., Batth, R.S., Kaur, S. and Gianini, G., 2021. Multi-disease big data analysis using beetle swarm optimization and an adaptive neuro-fuzzy inference system. *Neural Computing and Applications*, pp.1-12.
16. Shalaby, M., Belal, N.A. and Omar, Y., 2021. Data Clustering Improves Siamese Neural Networks Classification of Parkinson’s Disease. *Complexity*, *2021*.
17. “Telugu Character Dataset," [Online]. Available: http://lipitk.sourceforge.net/datasets/teluguchardata.htm
18. V. Babu, L. Prasanth, R. Sharma, G. V. Rao and A. Bharath, "HMM-Based Online Handwriting Recognition System for Telugu Symbols," in Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR), Chicago, 2007.
19. L. Prasanth, V. J. Babu, R. R. Sharma, G. V. P. Rao and M. Dinesh, "Elastic Matching of Online Handwritten Tamil and Telugu Scripts Using Local Features," in Proceedings of Ninth International Conference on Document Analysis and Recognition (ICDAR), Curitiba, Brazil, 2007.