International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

IJISAE

ISSN:2147-6799 www.ijisae.org Original Research Paper

A Novel Script-Rule-Based Character Segmentation Method for Devanagari Script in a Natural Scene Image

Vijay Prasad [1], Dr. Pranab Das [2], Dr. Y. Jayanta Singh [3]

Submitted: 28/12/2023 **Revised**: 04/02/2024 **Accepted**: 11/02/2024

Abstract: Devanagari script is one of the most popular and commonly utilized scripts across India. In this script, every character signifies a consonant sound along with an integral vowel sound. During the last decade, huge research is conducted for text localization and detection of various languages such as Japanese, English, Chinese, and others in scene images. However, there is limited research studies were performed on character segmentation for Devanagari Script. The existing methods used in the Devanagari script have some limits such as low accuracy in line, word, and character segmentation, dependency on handcrafted features in certain settings, computational complexity, and segmentation errors, etc. This research proposes a novel script-rule-based character segmentation model for the Devanagari script in a natural scene image setting. This proposed method is based on Deep Belief Network (DBN), Capsule Neural Network (CapsNet), and Naïve Bayes. Additionally, to recognize the middle region of the segmented region of the image, a modified recurrent neural network (RNN) model has been utilized. The measured performance metrics such as accuracy, precision, recall, and F1-score on Vpd Datasets using AdaDelta optimizer are 98.62%, 98,12%, 97.43%, and 97.01%, respectively. It is found that the proposed script rule-based character segmentation method obtains very optimized results.

Keywords: *Character Segmentation, CapsNet, Devanagari Script, DBN, Natural Scene Image, Naïve Bayes.*

1. Introduction

Devanagari script is one of the oldest writing methods which is utilized for multiple languages namely Hindi, Sindhi, Marathi, Sanskrit, and many others used primarily across East Asian countries. Devanagari script segmentation is one of the critical jobs particularly in natural scene images due to several reasons namely the complex structure of the characters, noise and artifacts presence, and altered writing patterns [1], [2]. Correct segmentation of the Devanagari script is essential for numerous applications namely image processing and optical character recognition. In the modern era, due to technological development, machine learning (ML), as well as deep learning (DL) methods, have revealed promising outcomes in numerous image processing application, which involves character segmentation and recognition [3], [4]. In the modern era, automated segmentation and recognition of the Devanagari script characters are ubiquitous. Due to the development in the area of artificial intelligence (AI) research, many character segmentation and recognition methods for different scripts have been explored which are based on numerous ML and DL algorithms [5], [6]. The automated character recognition of the Devanagari script is utilized in multifarious spheres of daily life namely reading the postal address, detection of the test present on the traffic signals, test recognition in the supermarkets, open parking, and many more [7].

The optical character recognition (OCR) system permits the Devanagari text identification from various natural scene digital pictures, scanned documents, etc. The OCR system evaluates the text image and transforms it into a machine-readable and further some other operations namely editing, searching, and indexing are done accordingly [7], [8]. OCR systems may be employed for multiple purposes namely scanning documents and images, automatic indexing, etc. The OCR software may be incorporated in a variety of systems namely management of documents, mobile apps, and workflow systems [9]-[11]. The conventional ML algorithms namely multilayer perception machines (MLP), random forest (RF), support vector machine (SVM) and many other uses the shallow architecture for handling the computing units as well as finite samples variety [12]. Devanagari script recognition and identification from the scene images is a complicated task owing to the low resolution of the images, rough layout, etc. For improving the image quality pragmatic pre-processing is very essential to attain a higher accuracy rate [13].

This research implementation is done using the newly formulated datasets namely the Vpd dataset which contains natural scene images. The contributions of the proposed research are given as follows:

[1] *Assistant Professor, Department of CA, SOT, Assam Don Bosco University, Guwahati, Assam, India.*
vpd.vijay82@gmail.com, Orcid Id- 10000-0003-1996-4910
[2] *HOD, Department of CA, SOT, Assam, Don Bosco University, Guwahati, Assam, India.*
pranab17@gmail.com, 0000-0003-0359-6615
[3] *Executive Director, NIELIT, Guwahati, Assam, India.*
yjayanta@gmail.com, 0000-0002-2886-1625

- To develop a novel script-rule-based character segmentation method for the Devanagari script in a natural scene image.
- This model is centered on a combination of the DBN, CapsNet, and Naïve Bayes algorithm to attain the optimized performance in comparison to the previous methods.
- To validate the proposed model on different optimizers namely the AdaDelta, SGD, and RMSprop to analyze the optimal performance of the model.
- To train and test the model in different hyperparameter settings for optimized results.

This research study proposes a novel script-rule-based character segmentation model for the Devanagari script in natural scene images. This proposed model is centered on a combination of the DBN, CapsNet, as well as Naïve Bayes algorithm. The model aimed to address the key problems posed by the Devanagari script segmentation of natural scene images.

2. Literature Review

This section explores various state-of-the-art methods for text identification, segmentation, and recognition through ML, DL, and other hybrid methods. In a study, S. Ahlawat *et al.* [14], proposed an enhanced handwritten digit identification utilizing the convolutional neural network (CNN). The conventional approaches for handwritten identification rely on different handcrafted features as well as huge previous information. OCR training rooted in such a prerequisite is a critical job. Another study by B. Al-Sadawi *et al.* [15], explored an extended-performance printed Arabic OCR based on an artificial neural network (ANN) classifier. This research investigates the Arabic OCR system to attain an enhanced accuracy rate. However, this method is limited to only the Arabic script and provides low accuracy in the line, word, and character segmentation. H. Butt *et al.* [16], discovered an attention mechanism based on CNN-RNN Arabic script text identification from the scene pictures. As per the reports, the Arabic language is spoken by approximately 422 million people. In this proposed model, datasets are applied as input, and features are generated utilizing CNN. Further, the extracted features sequence is translated to a bidirectional RNN to attain an ordered feature sequence and allows the developed model for selecting the appropriate data through the featured sequence. The attention mechanism is implemented by end-to-end iterative training as means of a standardized backpropagation algorithm. This approach has some common limits such as intricacy in character identification, limited generalization and minimal accuracy, etc.

M. Sonkusare *et al.* [17], conducted a review study based on the diverse previously explored segmentation research for the Devanagari script. In this study, several methods namely thresholding, dilation protocols, and many others are investigated and compared. Accurate character segmentation is very essential to attain the best accuracy of the model. this study provides an overview of the methods used for character segmentation. S. Gunna *et al.* [18], experimented to enhance the scene text identification for Indian script with the help of font diversity and a transfer learning approach. Indian scene text reading process is one the complicated tasks owing to the utilization of provincial vocabulary, varied script fonts as well as the size of the text. This research explores the key variance in Latin as well as Indian scene text identification systems. It is found that by using transfer learning the accuracy is improved on synthetic image datasets having the common background. However, this proposed method has some limits which include limited generalization, overfitting on less data, and many others. S. Bin Ahmed *et al.* [19], carried research to explore the previous approach used in the identification of the cursive text of Arabic script from the scene pictures. The latest research provides insight into the increasing interest of the researcher in text identification as well as recognition of multiple scene pictures. Correct recognition of the scene text is complicated owing to the varied font size and styles, illumination alteration, complex background, and many more. This study provides a thorough discussion of the existing scene text discovery as well as recognition approaches, and future research direction in the area of correct segmentation and identification of the cursive text in the scene pictures.

S. Bin Ahmed *et al.* [20], explore the new datasets for Arabic as well as English Scene text identification along with the assessment through the invariant feature extraction method. This study explores a new method by utilizing the adaptive maximal stable extremal zone method which determines the different scale-invariant features. This research provides more insights into the benchmark research in the area of scene text analysis. This research has some drawback which includes the limited robustness to variations, lacking model generalization, and high computational complexity. S. Y. Arafat *et al.* [21], explored a model for Urdu language text identification as well as recognition in multifarious scene pictures utilizing the DL approach. In this model, initially, the customized Faster RCN was utilized for the scene text localization and recognition. This approach has some limits which are hyperparameter sensitivity, low generalization, and more training time. H. Raj *et al.* [22], conducted a study for Devanagari script text identification from scene images. Scene pictures contain the data in text form and offer important insights for diverse applications rooted in image processing. In this work, a novel method for the Devanagari text identification is proposed which is based on the connected components and mathematical morphological operations.

3. Methodology

3.1. Dataset:

Datasets play a crucial role to train, testing, and improving the outcome of the novel models that are developed based on a combination of diverse methods. The dataset provides the model foundation for learning and making correct prediction outcomes. Dataset size, quality, and diversity significantly impact developed model generalization ability and prediction outcomes. In this research work, a novel Vpd Dataset is developed. This entire dataset has been pre-processed for improving the dataset's suitability for model training and testing. Character segmentation of diverse scripts is one of the crucial parts of the OCR procedure. For a better understanding of the natural scene image text, effective character segmentation is one of the top priorities.

3.2. Sample Size:

In this research, a script-rule-based character segmentation method is proposed for the Devanagari script in scene images. This section provides detailed insights into the proposed methodology for effective character segmentation of the Devanagari script. The details of the dataset used in this experimental work are depicted in Figure 1. For, effective segmentation accuracy measurement, the proposed Vpd dataset has been standardized utilizing multiple preprocessing techniques for the enhanced predictive outcome of this explored model. The total number of images, average number of words, total number of words, and number of letters used in the model implementation are 498, 14.4, 2812, and 13195, respectively.

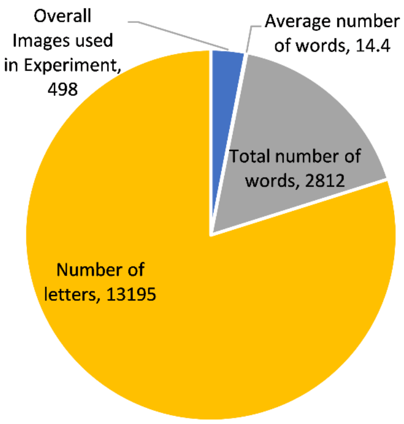

Fig 1: Details of the datasets used in implementation work.

3.3. System Configuration:

This proposed model has been implemented with a personal computer having the following system configuration: RAM: 16 GB DDR4, Video Card: NVIDIA GeForce RTX 30 Series, Windows 10, Processor: 12th Gen Intel i7. The experiment on the Vpd Dataset is conducted using the Jupiter notebook, which is an open-source web-based computing software. Python 3.11 version has been used for the programming. There have been used other modules in this experimental work namely Tensor Flow Kera's, Matplotlib, scikit-learn, etc.

3.4. Design:

The research design is carried out by utilizing a new customized dataset namely the VpdDataset. Figure 2 shows the proposed script-rule-based character segmentation model for the Devanagari script. Initially, the entire customized data is standardized utilizing enhanced preprocessing. The raw images used in the model training and testing may have a negative consequence on the model accuracy to determine the segmented character. A few of the common challenges are identified related to skew correction, thinning, etc. For improving the developed model accuracy in the training and testing phase, enhanced pre-processing is utilized. This method involves several operations namely binarization, skew correction, denoising, thinning, and upscaling. The binarization process alters the image into a binary format and represents every pixel of the image either white or black. This is one of the significant steps that improve the contrast between the background of the image and the characters. In this work, the binarization technique used in the OCR is based on adaptive thresholding. Further, skew-correction has been done to alter the rotated text angle of the image. Noise has a negative impact on the OCR performance, hence effective denoising aids in improving the image quality. The denoising process is done using the media filter. The thinning process minimizes the image's thickness by removing boundary pixels as well as maintaining the image's shape. The upscaling process is employed for improving the resolution of the images. In this model, random forest algorithms were used for the image's upscaling process. After dataset standardization using the enhanced pre-processing algorithms, segmentation of line words and characters is done for entire Vpd dataset images. Segmentation is one of the essential steps in the OCR as it performs the image separation in its constituent fragments i.e., lines, words, and characters. In the line segmentation process, every image of the dataset has been split into lines and processed in a line-by-line manner. In word segmentation, operation every line was split into words through spacing technique.

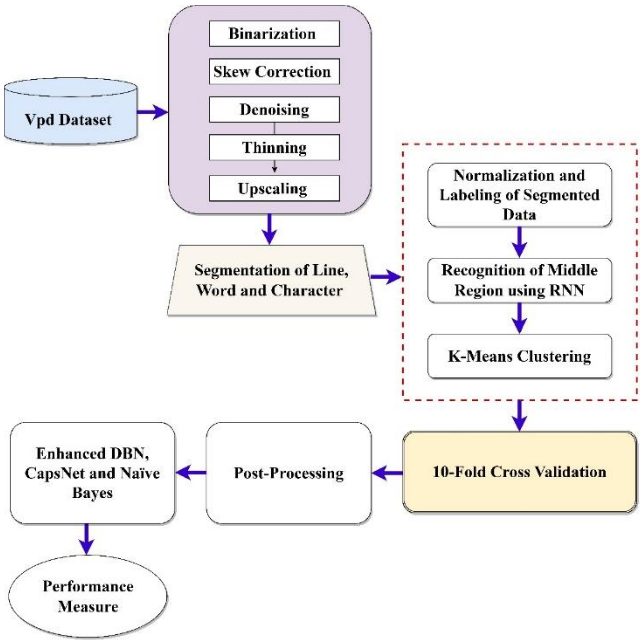

Fig 2: Proposed script-rule-based character segmentation model for Devanagari script.

Furthermore, in the character segmentation operation, the images containing the words were split into identical characters. In the next step, normalization and labeling of the entire dataset are done. The recognition process of the middle region is done utilizing the recurrent neural network (RNN) algorithm. For clustering of the segmented datasets, a widely recognized K-means clustering algorithm is opted for identifying the data object in obtained datasets for partition of n number of observations into the k number of clusters, wherein every observation belongs to a unique cluster having closest mean value or some cluster centroid. For performance assessment and generalization capability verification of the proposed model, a 10-fold cross-validation method has been used. This method aids in dividing the obtained datasets into two groups i.e., train and test into 80 and 20 ratios. Furthermore, post-processing is used for performing several operations namely spell check, confidence scoring, contextual evaluation, and language model. These techniques and models' effective utilization provides higher accuracy by a pragmatic margin. The developed model based on the enhanced DBN, CapsNet, and Naïve Bayes has been trained and tested using the obtained split datasets, and results are measured and validated.

Pseudo Code:

Input: VpD Dataset

Output: Vpd_A, Vpd_P, Vpd_R, Vpd_F.

Step 1: To collect the Datasets defined as VpDM.

Step 2: To initialize the defined dataset Vpd for preprocessing to obtain EVpD.

Step 3: Repeat the last step till EVpD in definite goal line GL.

Step 4: To begin segmentation of line word and character to obtain K_L, W_L, and C_L.

If,

K_L, W_L, and C_L are optimal to proceed further,

else

end the program and repeat the last step.

Step 5: To begin normalization and labeling of K_L, W_L, and C_L to obtain SK_L, SW_L, and SC_L.

Step 6: To initiate the middle region recognition of obtained SK_L, SW_L, and SC_L using RNN.

Step 7: if

SK_L, SW_L, and SC_L middle region are recognized optimally, proceed further

else

end the program and repeat the last step.

Step 8: To Initiate clustering operation over SK_L, SW_L, and SC_L.

Step 9: To begin cross-validation according to 10-Folds to split data into train and test groups.

Step 10: Initialize the received data for post-processing to refine the outcome.

Step 11: Initialize the system for training and testing the developed model based on DBN, CapsNet, and Naïve Bayes.

Step 12: if

Training and testing meet defined goal criteria and proceed further.

else,

end the program and repeat the last step.

Step 13: To evaluate the optimized parameters i.e., Vpd_A, Vpd_P, Vpd_R, Vpd_F,

3.5. Evaluation Metrics:

The performance of this new script-rule-based character segmentation model for the Devanagari script in a natural scene image is evaluated using the distinct evaluation metrics which involve accuracy, recall, precision, and F1-score.

The accuracy metric computes the predicted labels matching along with the original labels. For the optimal outcome of the model, accuracy is more desired enhanced. Accuracy is defined in Equation 1.

$$Accuracy = \frac{TP+TN}{FP+FN+TP+TN} \quad (1)$$

The recall metric is utilized for the evaluation of correct recognized labels. It involves total positive labels as well as forms that number of labels accurately identified. The recall equation is given in Equation 2.

$$Recall = \frac{TP}{FN+TP} \quad (2)$$

The precision metric evaluates the quality of the accurate outcome provided by the developed model. This relies on the predicted False Positive and True Positive. It is described in equation 3.

$$Precision = \frac{TP}{FP+TP} \quad (3)$$

The F1 score is utilized for assessing the model performance, specifically in the binary classification process. This groups the precision as well as recall in one source and offers a balanced measurement of model accuracy. It is given in equation 4.

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision+Recall} \quad (4)$$

In the above-defined evaluation metrics, TP defines the true positive values. The abbreviation TN is the True Negative. FP represents the False Positive values. The term FN represents the False Negative.

4. Results and Discussion

Devanagari script is a widely recognized Asian script primarily utilized across India in numerous languages involving Hindi, Sanskrit, and many others. Devanagari character segmentation is one of the complex procedures and a very challenging job owing to several reasons namely conjunct characters and modifier presence, overlapped strokes and smashed characters, alteration in fonts and writing patterns, degraded quality of the images, and many more. The aforementioned major issues make the Devanagari script character segmentation very intricate and complex. Hence, this research is aimed to provide a more robust and accurate method for Devanagari Script segmentation in natural scene images. The proposed script-rule-based character segmentation approach involves Deep Belief Network (DBN), Capsule Neural Network (CapsNet), and Naïve Bayes.

The details of the system configuration and programming environment for the implementation of the new script-rule-based model are given in Table 1. This script rule-based enhanced model implementation work is done on a personal computer having the given system configuration: Video Card: NVIDIA GeForce RTX 30 Series, RAM: 16 GB DDR4, Windows 10, and Selected Processor: 12th Gen Intel i7. The testing on the Vpd Dataset has been performed utilizing Jupiter Notebook. It is open-source web-based computing software, widely used by researchers for new model implantation and performance computation. Python 3.11 version has been utilized for coding purposes. A few of the other essential modules were used in this experimental work which involves Tensor Flow Kera's, Matplotlib, sci-kit-learn, etc.

Table 1: Details of system configuration and programming environment.

S. No.	System Arrangement	Details
1	Processor	12th Gen Intel i7
2	Frameworks	Tensor Flow Kera's, Matplotlib, sci-kit-learn
3	GPU (Graphic Processing Unit)	NVIDIA GeForce RTX 30 Series
4	RAM	16 GB DDR4
5	Programming Environment	Jupiter notebook, Python 3.11.3

Table 2: Hyperparameter specifics of proposed model.

S. No.	Hyperparameters	Details
1	Epochs	10
2	Batch-Size	15
3	Rate of Learning	0.0002
4	Filters	15
5	Optimizer	Adadelta, RMSprop, and SGD
6	Activation function	ReLU (Rectified Linear Unit)
7	Dropout value	0.4

The hyperparameter selection plays a crucial role in obtaining the optimized outcome in model training and validation. The hyperparameters aid in controlling the overall training and testing process and evaluate the model parameter values in real-time. Hyperparameter selection maximizes the model performance, provides more stability, aids in generalization, and improves the model interpretability and efficiency in a significant and optimal manner. Table 2 describes the used hyperparameter specifics proposed for the Devanagari text segmentation. The selected hyperparameters description is as follows: number of epochs, batch size, rate of learning, filters, and dropout values are 10, 15, 0.0002, 15, and 0.4, respectively. The ReLU (Rectified Linear Unit) activation function has been selected in the proposed model training. In addition to this, the proposed model performance was computed utilizing the three different optimizers namely the Adadelta, RMSprop, and SGD, respectively, for optimal validation of the model.

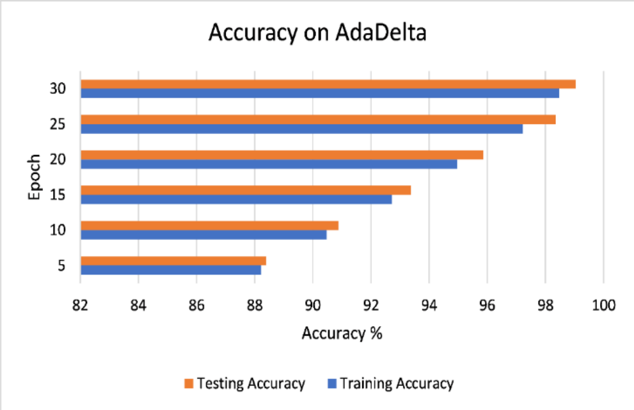

Fig 3: Measured accuracy on AdaDelta optimizer.

Figure 3 presents the measured accuracy of the AdaDelta optimizer. The accuracy metric is one of the primary criteria to assess the performance of the model. While the AdaDelta optimizer was selected in the model training, the computed training accuracy on the epochs 5, 10, 15, 20, 25, a 30 is observed at 88.22%, 90.47%, 92.72%, 94.97%, 97.22%, and 98.47%, respectively. Also, utilizing the AdaDelta optimizer in the testing, the computed testing accuracy on the epochs 5, 10, 15, 20, 25, and 30 is observed at 88.39%, 90.88%, 93.37%, 95.86%, 98.35%, and 99.04% respectively.

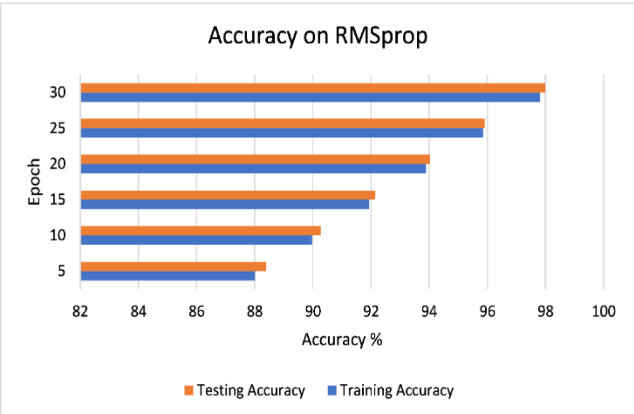

Fig 4: Measured accuracy on RMSprop optimizer.

Figure 4 presents the measured accuracy of the RMSprop optimizer. While the RMSprop optimizer has been selected in the model training, the computed training accuracy on the epochs 5, 10, 15, 20, 25, a 30 is observed at 88.01%, 89.97%, 91.93%, 93.89%, 95.85%, and 97.81%, respectively. Further, utilizing RMSprop optimizer in testing, the computed testing accuracy on the epochs 5, 10, 15, 20, 25, and 30 is observed 88.39%, 90.27%, 92.15%, 94.03%, 95.91%, and 97.99%, respectively.

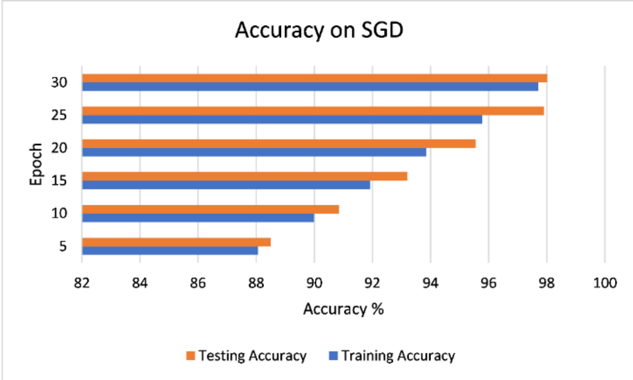

Fig 5: Measured accuracy on SGD optimizer.

Figure 5 depicts the measured accuracy of the SGD optimizer. While SGD optimizer is selected in the model training, the computed training accuracy on the epochs 5, 10, 15, 20, 25, and 30 is observed at 88.06%, 89.99%, 91.92%, 93.85%, 95.78%, and 97.71%, respectively. Moreover, utilizing SGD optimizer in testing, the computed testing accuracy on the epochs 5, 10, 15, 20, 25, and 30 is observed at 88.5%, 90.85%, 93.2%, 95.55%, 97.9%, and 98.02%, respectively.

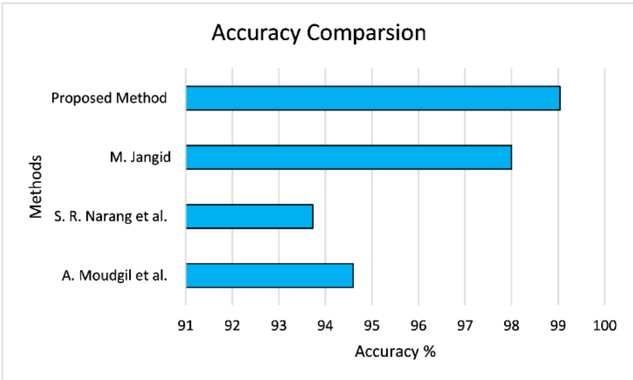

Fig 6: Accuracy comparison of new script-rule-based model and existing methods [23]–[25].

Figure 6 depicts a comparison of the new script-rule-based model and existing methods. For the comparative evaluation of this new script-based model with previous techniques, AdaDelta optimizer testing accuracy is considered highly improved over other RMSprop and SGD optimizers. The obtained maximal accuracy of the previous model by M. Jangid *et al.*, S.R. Narang *et al.* and A. Moudgil *et al.* was 94.6%, 93.73%, and 98%, respectively. The proposed model archives higher accuracy of 99.04% in comparison with previous models. Therefore, it is apparent from the accuracy comparative analysis that the new script-rule-based proposed model attains a higher accuracy rate.

Table 3: Illustrates measured precision, recall, and F1-Score of new script-rule-based models on AdaDelta optimizer.

S. No.	Performance Metric	Classifier	Value in (%)
1	Precision	DBN, CapsNet, Naïve Bayes	98.12%
2	Recall	DBN, CapsNet, Naïve Bayes	97.43%
3	F1-Score	DBN, CapsNet, Naïve Bayes	97.01%

Table 3 shows an analysis of measured precision, recall, and F1-Score of proposed models on the AdaDelta optimizer. The selected classifier model is based on the DBN, CapsNet, and Naïve Bayes. The observed values of the precision, recall, and F1-score using the AdaDelta optimizer are 98.12%, 97.43%, and 97.01%, respectively. Therefore, it is observed that using the AdaDelta optimizer on Vpd Dataset provides pragmatic outcomes for all the performance metrics.

Table 4: Analysis of measured train-test losses.

Optimizer Used	Losses	Epochs					
		5	**10**	**15**	**20**	**25**	**30**
AdaDelta	Training Loss	3.4	2.6	2.1	1.4	0.7	0.4
	Testing Loss	3.2	2.3	1.9	1.3	0.6	0.3
RMSprop	Training Loss	3.9	3.2	2.7	2.1	1.8	1.2
	Testing Loss	3.7	3.0	2.3	1.9	1.7	1.1
SGD	Training Loss	3.6	3.1	2.8	2.7	2.1	1.8
	Testing Loss	3.4	3.0	2.5	2.1	1.9	1.6

Table 4 shows an analysis of measured train-test losses. Training loss metrics evaluate how well the model fits train datasets. This quantifies the discrepancy between the proposed model forecasts as well as the real goal values in training. One of the key goals of the training procedure is to reduce this error rate through alteration in model parameters which involve weights as well as biases. The testing loss or error rate aids in assessing how well the model predicts and generalizes novel unseen instances. For better performance and generalization of model capability, a minimal error rate is desired. In this experiment, the error rate is recorded in the training and testing phases using different optimizers. From the train test error rate is observed that the AdaDelta optimizers attain minimal error rate and high performance over the other two optimizers i.e., RMSprop and SGD. On the AdaDelta optimizer, training and testing loss measured on epochs 5, 10, 15, 20, 25, and 30, are 3.4%, 2.6%, 2.1%, 1.4%, 0.7%, 0.4%, and 3.2%, 2.3%, 1.9%, 1.3%, 0.6%, 0.3%, respectively. On the RMSprop optimizer, training and testing loss measured on epochs 5, 10, 15, 20, 25, and 30, are 3.9%, 3.2%, 2.7%, 2.1%, 1.8%, 1.2%, and 3.7%, 3.0%, 2.3%, 1.9%, 1.7%, 1.1%, respectively. Lastly On the SGD optimizer, training and testing loss measured on epochs 5, 10, 15, 20, 25, and 30, are 3.6%, 3.1%, 2.8%, 2.7%, 2.1%, 1.8%, and 3.4%, 3.0%, 2.5%, 2.1%, 1.9%, 1.6%, respectively.

Table 5: Analysis of Probability Distribution of Dataset for Different Optimizers according to subject wise.

Optimizers	Subjects					
	1	**2**	**3**	**4**	**5**	**6**
RMSprop	0.004	0.044	0.451	0.006	0.341	0.009
SGD	0.003	0.001	0.234	0.005	0.372	0.034
AdaDelta	0.6750	0.008	0.345	0.012	0.045	0.034

After post-processing of the suggested model centered on the DBN, CapsNet, and Naïve Bayes, analyzed the data with six diverse subject areas according to the provided Vpd dataset for all the selected optimizers. These different six subjects were labeled as Subject 1, Subject 2, Subject 3, Subject 4, Subject 5, and Subject 6. In this experimental work, Subject 1 defines the natural scene images i.e., road, tourism, railway platforms etc. Subject 2 includes the image data related to secured zones i.e., National parks, museums, etc. Subject 3 involves the images data related to the public place's town squares, urban farms, etc. Subject 4 includes the image data related to the marketplace i.e., supermarkets, general stores etc. Subject 5 covers the image data related to the hill areas i.e., private or open parking, river sides, etc. Lastly, subject area 6 includes the colleges and university campuses, for instance, libraries, cafeterias, etc. Table 5 represents the analysis of probability distribution for different segments subject-wise. The probability distribution of Subject 1, Subject 2, Subject 3, Subject 4, Subject 5, and Subject 6, for RMSprop optimizer was 0.004, 0.044, 0.451, 0.006, 0.341, and 0.009. The probability distribution of Subject 1, Subject 2, Subject 3, Subject 4, Subject 5, and Subject 6, for SGD optimizer was 0.003, 0.001, 0.234, 0.005, 0.372, and 0.034. The probability distribution of Subject 1, Subject 2, Subject 3, Subject 4, Subject 5, and Subject 6, for the AdaDelta optimizer was 0.6750, 0.008, 0.345, 0.012, 0.045, and 0.034.

5. Conclusion

The Devanagari script character segmentation of scene images is very challenging owing to the different orientations of the lines, words, and characters, low-resolution images, variation in the writing patterns, etc. The automated segmentation and recognition of the Devanagari script is getting massive attention from research nowadays because of development in image processing as well as the computer vision field. This study investigates a novel script-rule-based character segmentation method for the Devanagari script in a natural scene image, which is built using a combination of the DBN, CapsNet, and Naïve Bayes algorithm. This developed model has been trained and validated using altered hyperparameter settings and different optimizers namely the AdaDelta, SGD, and RMSprop for optimal validation and performance assessment. The evaluation metrics such as accuracy, precision, recall, and F1-score on Vpd Datasets using AdaDelta optimizer are 98.62%, 98,12%, 97.43%, and 97.01%, respectively. After a comparative analysis of the measured results on different optimizers, it is evident that AdaDelta attains the optimal performance metrics. Furthermore, another key advantage of this work is that it attains a very minimal error rate on the AdaDelta optimizer which makes the proposed model more suitable for the natural scene image segmentation and recognition purpose. There is a vital future research scope on Devanagari text discovery and recognition in natural scene pictures. This research may be extended for dataset creation and augmentation, handling of overlapping and broken characters, etc.

References:

[1] D. Cao, Y. Zhong, L. Wang, Y. He, and J. Dang, “Scene Text Detection in Natural Images: A Review,” *Symmetry (Basel).*, vol. 12, no. 12, p. 1956, Nov. 2020, doi: 10.3390/sym12121956.

[2] V. Naosekpam and N. Sahu, “Text detection, recognition, and script identification in natural scene images: A Review,” *Int. J. Multimed. Inf. Retr.*, 2022, doi: 10.1007/s13735-022-00243-8.

[3] N. Gupta and A. S. Jalal, “Traditional to transfer learning progression on scene text detection and recognition: A survey,” *Artif. Intell. Rev.*, 2022, doi: 10.1007/s10462-021-10091-3.

[4] T. Khan, R. Sarkar, and A. F. Mollah, “Deep learning approaches to scene text detection: a comprehensive review,” *Artif. Intell. Rev.*, 2021, doi: 10.1007/s10462-020-09930-6.

[5] A. Khalil, M. Jarrah, M. Al-Ayyoub, and Y. Jararweh, “Text detection and script identification in natural scene images using deep learning,” *Comput. Electr. Eng.*, 2021, doi: 10.1016/j. compeleceng.2021.107043.

[6] M. Ibrayim, Y. Li, and A. Hamdulla, “Scene Text Detection Based on Two-Branch Feature Extraction,” *Sensors*, 2022, doi: 10.3390/s22166262.

[7] D. Pandey, B. K. Pandey, and S. Wairya, “Hybrid deep neural network with adaptive galactic swarm optimization for text extraction from scene images,” *Soft Comput.*, 2021, doi: 10.1007/s00500-020-05245-4.

[8] K. Boukthir, A. M. Qahtani, O. Almutiry, H. Dhahri, and A. M. Alimi, “Reduced annotation based on deep active learning for arabic text detection in natural scene images,” *Pattern Recognit. Lett.*, 2022, doi: 10.1016/j.patrec.2022.03.016.

[9] D. Zhong, P. Shivakumara, L. Nandanwar, U. Pal, M. Blumenstein, and Y. Lu, “Local Resultant Gradient Vector Difference and Inpainting for 3D Text Detection in the Wild,” *Int. J. Pattern Recognit. Artif. Intell.*, 2022, doi: 10.1142/S0218001422530056.

[10] A. Ali, M. Pickering, and K. Shafi, “Urdu Natural Scene Character Recognition using Convolutional Neural Networks,” 2018. doi: 10.1109/ASAR.2018.8480202.

[11] O. Y. Ling, L. B. Theng, A. C. Weiyen, and M. Christopher, “Development of Vertical Text Interpreter for Natural Scene Images,” *IEEE Access*, 2021, doi: 10.1109/ACCESS.2021. 3121608.

[12] A. A. A. Ali and S. Mallaiah, “Intelligent handwritten recognition using hybrid CNN architectures based-SVM classifier with dropout,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 6, pp. 3294–3300, 2022, doi: 10.1016/j.jksuci.2021.01.012.

[13] M. S. V. Dharsini, N. Kamath, Y. Chaudhary, A. Sriramoju, and N. Anurag, “Devanagri Character Image Recognition and Conversion into Text using Long Short Term Memory,” 2022. doi: 10.1109/ICDSAAI55433.2022.10028963.

[14] S. Ahlawat, A. Choudhary, A. Nayyar, S. Singh, and B. Yoon, “Improved handwritten digit recognition using convolutional neural networks (Cnn),” *Sensors (Switzerland)*, 2020, doi: 10.3390/s20123344.

[15] B. Al-Sadawi, A. Hussain, and N. S. Ali, “High-Performance Printed Arabic Optical Character Recognition System Using ANN Classifier,” 2021. doi: 10.1109/PICICT53635.2021.00013.

[16] H. Butt, M. R. Raza, M. J. Ramzan, M. J. Ali, and M. Haris, “Attention-Based CNN-RNN Arabic Text Recognition from Natural Scene Images,” *Forecasting*, 2021, doi: 10.3390/forecast3030033.

[17] M. Sonkusare, R. Gupta, and A. Moghe, “A Review on Character Segmentation Approach for Devanagari Script,” 2021. doi: 10.1007/978-981-16-2248-9_19.

[18] S. Gunna, R. Saluja, and C. V. Jawahar, “Improving Scene Text Recognition for Indian Languages with Transfer Learning and Font Diversity,” *J. Imaging*, 2022, doi: 10.3390/jimaging8040086.

[19] S. Bin Ahmed, S. Naz, M. I. Razzak, and R. Yusof, “Arabic cursive text recognition from natural scene images,” *Appl. Sci.*, 2019, doi: 10.3390/app9020236.

[20] S. Bin Ahmed, S. Naz, M. I. Razzak, and R. B. Yusof, “A Novel Dataset for English-Arabic Scene Text Recognition (EASTR)-42K and Its Evaluation Using Invariant Feature Extraction on Detected Extremal Regions,” *IEEE Access*, 2019, doi: 10.1109/ACCESS.
2019.2895876.

[21] S. Y. Arafat and M. J. Iqbal, “Urdu-Text Detection and Recognition in Natural Scene Images Using Deep Learning,” *IEEE Access*, 2020, doi: 10.1109/ACCESS.2020.2994214.

[22] H. Raj and R. Ghosh, “Devanagari text extraction from natural scene images,” in *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Sep. 2014, pp. 513–517. doi: 10.1109/ICACCI.2014.6968472.

[23] M. Jangid and S. Srivastava, “Handwritten Devanagari character recognition using layer-wise training of deep convolutional neural networks and adaptive gradient methods,” *J. Imaging*, 2018, doi: 10.3390/jimaging4020041.

[24] A. Moudgil, S. Singh, V. Gautam, S. Rani, and S. H. Shah, “Handwritten devanagari manuscript characters recognition using capsnet,” *Int. J. Cogn. Comput. Eng.*, 2023, doi: 10.1016/j.ijcce.
2023.02.001.

[25] S. R. Narang, M. Kumar, and M. K. Jindal, “DeepNetDevanagari: a deep learning model for Devanagari ancient character recognition,” *Multimed. Tools Appl.*, 2021, doi: 10.1007/s11042-021-10775-6.