Indonesian Journal of Electrical Engineering and Computer Science
Vol. 36, No. 3, December 2024, pp. 1564~1572
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v36.i3.pp1564-1572

Efficient deep learning models for Telugu handwritten text recognition

Buddaraju Revathi[1], B. N. V. Narasimha Raju[2], Boddu L. V. Siva Rama Krishna[3]
Ajay Dilip Kumar Marapatla[3], S. Suryanarayanaraju[4]
[1]Department of Electronics and Communication Engineering, SRKR Engineering College, Bhimavaram, India
[2]Department of Computer Science and Engineering, SRKR Engineering College, Bhimavaram, India
[3]Department of Computer Science and Engineering, SRM University-AP, Mangalagiri, India
[4]Department of Information Technology, SRKR Engineering College, Bhimavaram, India

Article Info

Article history:

Received Apr 3, 2024
Revised Sep 2, 2024
Accepted Sep 7, 2024

Keywords:

Canny detection
Character segmentation
Convolutional neural network
Feature extraction
ResNet

ABSTRACT

Optical character recognition (OCR) technology is indispensable for converting and analyzing text from various sources into a format that is editable and searchable. Telugu handwriting presents notable challenges due to the resemblance of characters, the extensive character set, and the need to segment overlapping characters. To segment the overlapping characters, we assess the width of small characters within a word and segment the overlapping characters accordingly. This method is well suited for the segmentation of overlapping compound characters. To address the recognition of similar characters with less training periods we have used ResNet 18 and SqueezeNet models which have achieved character recognition rates of 95% and 94% respectively.

Corresponding Author:

Buddaraju Revathi
Department of Electronics and Communication Engineering, SRKR Engineering College
Bhimavaram, 534204, India
Email: buddaraju.revathi@gmail.com

1. INTRODUCTION

Optical character recognition (OCR) is a technology engineered to convert various document formats into text data that is editable and searchable [1]. This involves analyzing the textual images, recognizing the characters, and converting them into a machine-readable format. OCR technology plays a pivotal role in expediting the digitalization journey by transforming physical documents into formats that are searchable, editable, and machine-readable. This reveals an excess of advantages for businesses, institutions, and individuals alike.

OCR comprises five primary stages. Initially, image acquisition entails capturing the text-containing image, emphasizing high-quality acquisition for precise OCR results [2]. Subsequently, preprocessing improves image quality through techniques like noise reduction, contrast enhancement, and binarization [3]. The third phase, text detection and segmentation, employs algorithms to identify text regions in the pre-processed image and distinguish them from non-text elements. Then, individual characters or words within detected text regions are isolated to streamline recognition for the OCR system [4]. In the fourth stage, feature extraction captures pertinent text characteristics such as shape and size for recognition. Lastly, character recognition matches extracted features with predefined models or templates, utilizing pattern recognition or machine learning techniques to identify characters.

OCR for native languages requires heightened attention to preserve historical documents, facilitate automation, and support educational endeavors [5]. Despite being a prominent Indian language, Telugu has received less emphasis than others due to its intricate characters, segmentation challenges, and the lack of suitable datasets. Overcoming segmentation obstacles in Telugu mandates innovative strategies, like isolating the vattus, to improve OCR precision and utility.

Development of OCR for native languages should account for the language specific features for segmentation and recognition. To overcome segmentation in native languages Naveena and Aradhya [6] has developed an advanced algorithm for segmenting handwritten Kannada scripts. It combines thinning methods, branch point identification, and mixture models, utilizing the EM algorithm to learn complex Gaussian mixtures. Through strategic use of cluster mean points and branch points, Naveena achieves precise character segmentation, with an impressive accuracy of 85.5% demonstrated in experimental trials. Babu *et al.* [7] introduces a syllable segmentation method for printed Telugu text. It's crucial for tasks like word segmentation, speech synthesis, and information retrieval. The algorithm is inspired by Telugu script's structure but faces challenges due to language ambiguities. It proposes a non-dictionary approach based on syllable width, comparing it with the aspect ratio method. Experimental results demonstrate over 99% segmentation accuracy. Banumathi and Chandra [8] introduced a segmentation system, based on the Projection Profile method, handles general and historical handwritten text. It includes pre-processing steps to enhance image quality and achieves a 90% success rate in line segmentation. However, word segmentation, especially for Kannada script, poses challenges due to overlapping subscripts and unclear spacing. Despite these obstacles, the system represents a significant advancement in handwritten text segmentation, highlighting areas for improvement in Kannada word segmentation.

To deal with recognition of large number of similar characters Rani *et al.* [9] presented a model for recognizing handwritten Kannada characters, utilizing transfer learning from a Devanagari recognition system. The model employs a VGG19 NET architecture, comprising five blocks with convolution and max-pooling layers. Training involves 123,654 data samples, with experimental evaluation showing close to 90% accuracy. Validation over 10 epochs achieves 73.51% accuracy, demonstrating the effectiveness of transfer learning for Kannada character recognition. Sanjeev and Samuel [10] has developed an OCR system tailored for recognizing basic characters in printed Kannada text, accommodating various font sizes and types. By utilizing Hu's invariant moments and Zernike moments for feature extraction and neural classifiers for character classification, the system achieves a remarkable recognition rate of 96.8%. This methodology shows promise for extending recognition to other South Indian languages, particularly Telugu, demonstrating efficacy in accurately recognizing printed Kannada characters and indicating potential applicability to other languages in the region. Basha *et al.* [11] introduced a groundbreaking method that harnesses Deep CNNs to accurately identify handwritten Telugu alphabets depicted in images. Through extensive experimentation, Basha’s model achieves remarkable advancements, achieving an accuracy rate ranging between 80% to 95% in accurately recognizing individual alphabets.

Segmentation solutions for native language recognition need to address the specific traits of each language. However, advancements have been limited due to the difficulties in segmenting overlapping handwritten characters. To tackle this issue, improvements have been made to the vertical projection profiles algorithm, resulting in more precise segmentation of overlapping characters. Previously, researchers predominantly used machine learning or deep learning models for character recognition [12]-[14]. However, machine learning models often struggled to achieve high accuracy, and deep learning models generally required many parameters, leading to significant computational challenges.

2. OCR TEXT DETECTION

2.1. Text detection for the test page

At first, the OCR system takes the scanned page with the text to be recognized as input. Following this, the system converts the image to grayscale. Afterward, the bilateral filter, known as a non-linear filter, smooths an image while preserving its edges. If M is scanned image and a is the pixel location, then filtered output of bilateral filter $BF(M)(a)$is given by (1).

$$BF(M)(a) = \frac{1}{W(a)} \sum_{b \epsilon R} M(b) \cdot K_{\sigma_r} (\|a-b\|) \cdot K_{\sigma_{rg}} (\|M(a)-M(b)\| \quad (1)$$

where R is the spatial neighborhood of a and $K_{\sigma_r}$is the spatial Gaussian kernel with standard deviation σ_r,$K_{\sigma_{rg}}$is the range Gaussian kernel with standard deviation $\sigma_{rg}.w(a)$ is the normalization factor.

Sobel edge detection aids in extracting edges from an image, facilitating the recognition of object boundaries such as words [15]. Applying a threshold to the gradient magnitude image *M (a, b)*, produces a binary edge map, simplifying detection given by (2).

$$M_{bin} = \begin{cases} 1, & M(a,b) > T \\ 0, & elsewhere \end{cases} \quad (2)$$

where M_{bin} is binarized image and *T* is the threshold level. Next, connected component analysis is utilized on the binary edge map to separate individual words, grouping connected pixels together. Once these components are obtained, intersection and union operations refine word detection by improving the delineation of word boundaries.

2.2. Segmentation and feature extraction

Character segmentation is indispensable for the accuracy and efficacy of OCR [16], [17]. It guarantees the isolation of individual characters, adeptly manages handwritten text, enhances recognition accuracy, accommodates diverse languages and scripts, boosts processing efficiency, and aids in document comprehension. Proper segmentation holds a pivotal position in achieving elevated recognition rates within OCR systems. The proposed character segmentation algorithm steps are given below. Figure 1 shows the character segmentation projection profile graphs and the segments. Figure 1(a) shows the projection profile graph for an input word whereas Figures 1(b) and 1(c) shows the segmented part obtained after applying standard algorithm and the proposed one.
Step 1: Begin with an input image $M(a,b)$and calculate the column-wise sum of pixel intensities [18] given by the (3):

$$D_a = \sum_{b=1}^{N} M_{ab} \quad (3)$$

Step2: Define segments where the same value repeats consecutively across columns using (4):

$$Segment = \{(a,h) | M_{ab} = M_{(b(a+1)}\} \quad (4)$$

where *h* is width of the part after segmentation.
Step3: Determine the minimum segment width S_{min} from the identified segments given in (5):

$$S_{min} = \min(h) \quad (5)$$

Step4: Compare the minimum segment width S_{min} with the widths of previously segmented blocks using (6):

$$h > 2 \times S_{min} \quad (6)$$

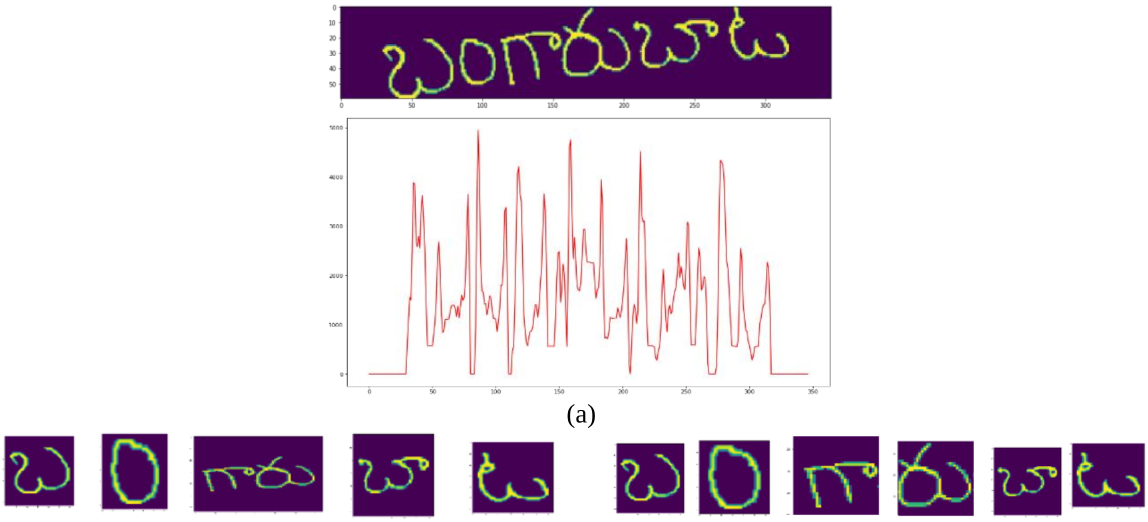
For segment width that exceeds double the minimum width, split the image at the column with the lowest pixel intensity from the middle, considering 10 columns on either side.

CNNs are preferred for extracting image features due to their ability to learn hierarchical features, display translation invariance, efficiently share parameters, employ local receptive fields, integrate pooling layers, maintain robustness to variations, and utilize pretrained models [19], [20]. Telugu language has intricated structures with many numbers of characters with similar structure. As the CNN architectures are capable to extract the features, we have used light weighted architectures to recognize Telugu characters.

The convolutional layer processes input images by convolving them with learnable filters to extract features. Each filter scans across the input image, producing feature maps through convolution operations, which are then passed through an activation function. Given an input image *X* with dimensions $h \times w \times c$, where *h* represents the height, *w* denotes the width, and *c* stands for the number of channels. If *A (a, b, z)* is pixel intensity of input image at *(a, b)* in channel z and $Q_f(a,b)$is the feature map at *(a, b)* given by (7):

$$Q_f(a,b) = \sigma(\sum_{z=1}^{Z} \sum_{da=1}^{Q} \sum_{db=1}^{Q} A(a+da, b+db, z). w_a(da, dy, z) + bias_b) \quad (7)$$

where $w_a(da, dy, z)$ represents the weight of filter and σ is the activation function. We have used Rectified Linear unit to activate the CNN model.



(a) (b) (c)

Figure 1. Character segmentation: (a) vertical projection profile graph for a word, (b) projection-based segmentation output, and (c) output of proposed segmentation

Max-pooling layer reduces the spatial dimensions of feature maps by selecting the maximum value within specified regions. This method captures important features while down sampling the data, thereby decreasing its resolution. If P is the pool size, then for feature map $at\ Q_f(a, b)$, the maximum value in the pooling region is given by (8).

$$MaxPool(Q_f)_{a,b} = max_{da,db}(Q_f(a.P + da, b.P + db) \quad (8)$$

The Adam optimizer is used to adjust the network's weights based on the loss function gradient. It combines aspects of the AdaGrad and RMSProp optimizers to adaptively update learning rates for each parameter during training. Later SoftMax transforms the output of the final layer into a probability distribution across multiple classes. It exponentiates the logits of each class and then normalizes them, producing probabilities. This normalization enables classification decisions based on the class with the highest probability.

After character segmentation, the segmented characters obtained are extracted for features. A CNN model is employed for feature extraction, trained using the IEEE dataport character dataset. We have used SqueezeNet and ResNet 18 models for feature extraction. Subsequently, the features extracted from the test segments are compared with those obtained during training. The class with the highest matching probability is then assigned as the output.

Previous research recommended employing machine learning models and very deep learning models for Telugu text detection. However, machine learning models yielded lower recognition rates, while deep learning models achieved higher recognition rates at the cost of increased trainable parameters and longer training periods [13], [14], [21]. To address these challenges, we adopted ResNet-18 and SqueezeNet models with variations in filter sizes. These models offer reduced parameter consumption while maintaining significant accuracy rates. The selection between these models depends on specific requirements and application contexts. The model of ResNet 18 is shown in Figure 2 and SqueezeNet is shown in Figure 3.

ResNet-18 holds significance for its ability to find equilibrium between depth, complexity, and performance, rendering it a flexible and efficient solution for diverse computer vision applications [22]. Its pioneering use of residual connections has reshaped the landscape of deep learning architectures, laying the foundation for the emergence of increasingly sophisticated models. While SqueezeNet stands out as a deep learning model tailored for heightened accuracy while conserving computational resources [23]. Its distinguishing feature is the incorporation of 1x1 convolutional layers, known as network-in-network, which efficiently reduces parameter count while preserving representational strength. This innovative approach empowers SqueezeNet to maintain high accuracy levels despite its notably smaller model size when compared with established deep architectures like VGG. Both ResNet18 and SqueezeNet are unique in architecture by providing optimal performance in image recognition tasks by utilizing fewer parameters compared to VGG.

Both models were trained using the IEEE Dataport dataset and assessed for their character recognition capabilities. Subsequently, a test page containing handwritten Telugu words was inputted into the OCR system. The OCR identified the words and segmented them into characters, which were then passed to the models for recognition. Based on the output, the system's word recognition rates were evaluated. The models were tested with 1000 words, and their performance was assessed accordingly.

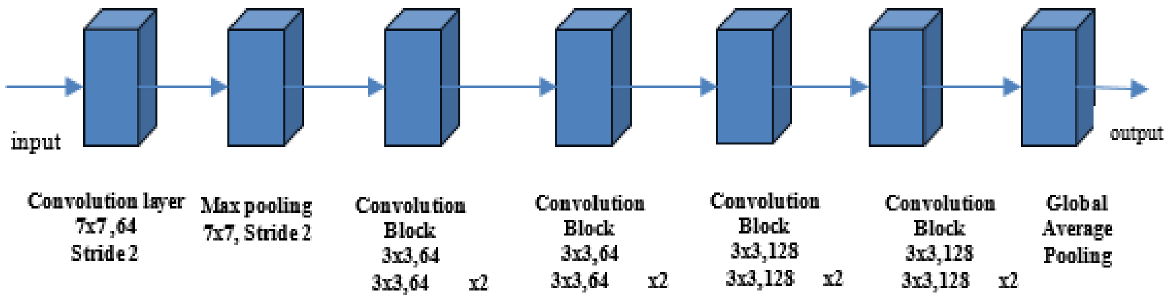


Figure 2. Model of ResNet 18

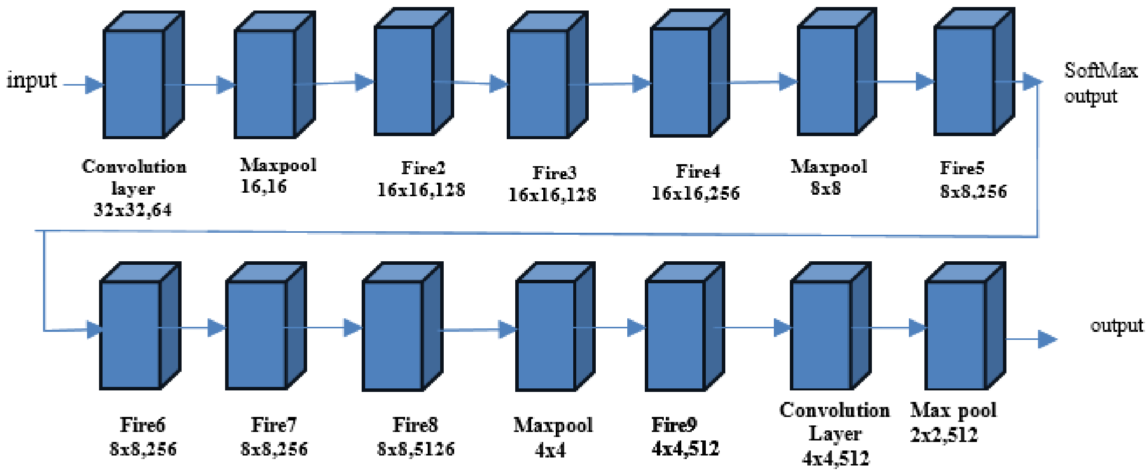


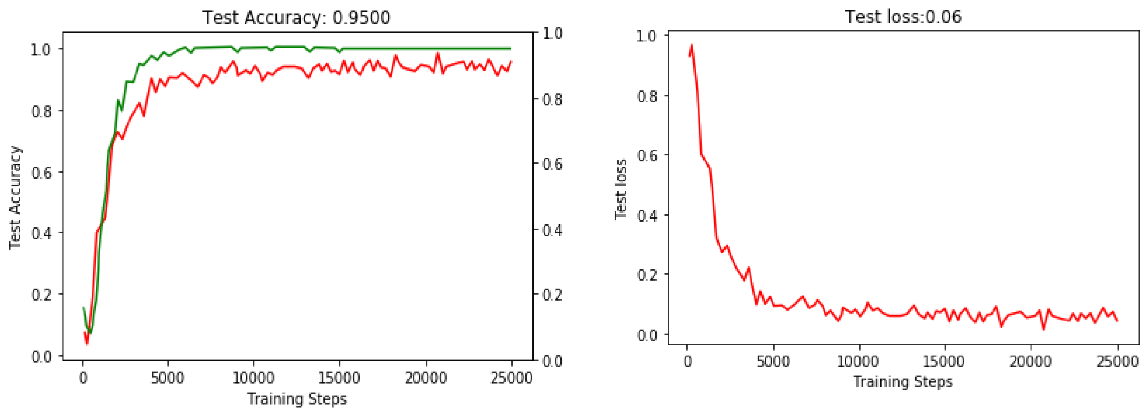
Figure 3. Model of SqueezeNet

3. RESULTS

Each image was converted to grayscale and resized to dimensions of 64x64 pixels. The dataset was obtained from IEEE Dataport [24], and training involved both ShuffleNet and ResNet 18 models over 25,000 steps. Training data comprised 11,602-character images, with an additional 2,565-character images reserved for validation purposes. Figure 4 shows the plots of ResNet18. Figure 4(a) displays the character recognition train and validation accuracy plots for ResNet18. Meanwhile, Figure 4(b) illustrates the validation loss plot of ResNet 18. Figure 5 shows the plots of SquuezeNet. Figure 5(a) depicts the character recognition training and validation accuracy plots for SqueezeNet, while Figure 5(b) showcases the validation loss plot of SqueezeNet.

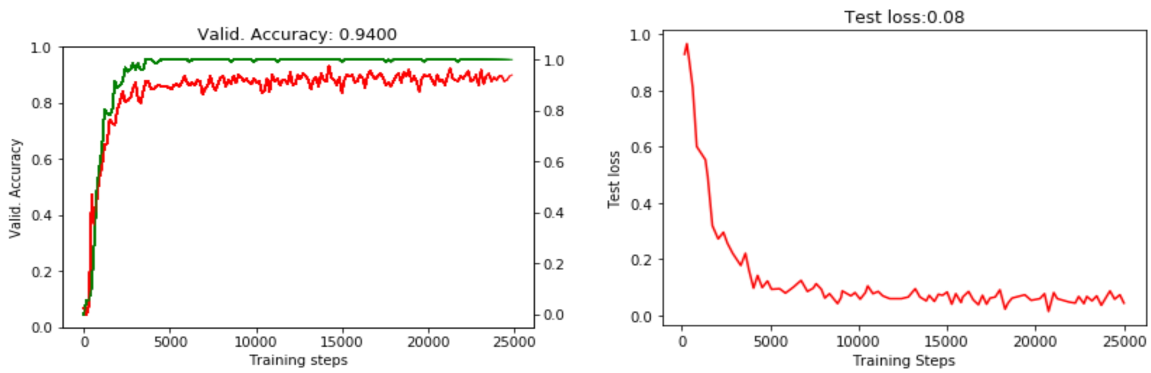
ResNet18 has reached a character accuracy of 95% and a loss of 6% based on the plots. Meanwhile, SqueezeNet achieves a character recognition rate of 94% with a loss of 8%, as indicated by the plot data. Both ResNet18 and SqueezeNet demonstrate noteworthy performance in recognizing Handwritten Telugu characters, showcasing substantial character recognition rates.

Table 1 presents a comparison between SqueezeNet and ResNet18 models. ResNet18, renowned for its architecture, requires 1,368,640 parameters, whereas SqueezeNet demands only 861,344 parameters. Despite having fewer trainable parameters, SqueezeNet achieves accuracy rates closer to ResNet18. The validation, test, and word accuracy rates for ResNet18 are 95%, 90.58%, and 84%, respectively. On the other hand, SqueezeNet achieves a validation accuracy of 94%, a test accuracy of 89.438%, and a word accuracy of 83.5%. Both models demonstrate strong performance on a Telugu character dataset. The choice between them depends on the specific problem requirements and desired accuracy rates.



(a) (b)

Figure 4. Plots of ResNet18 (a) accuracy plot and (b) loss plot



(a) (b)

Figure 5. Plots of SqueezeNet (a) accuracy plot and (b) loss plot

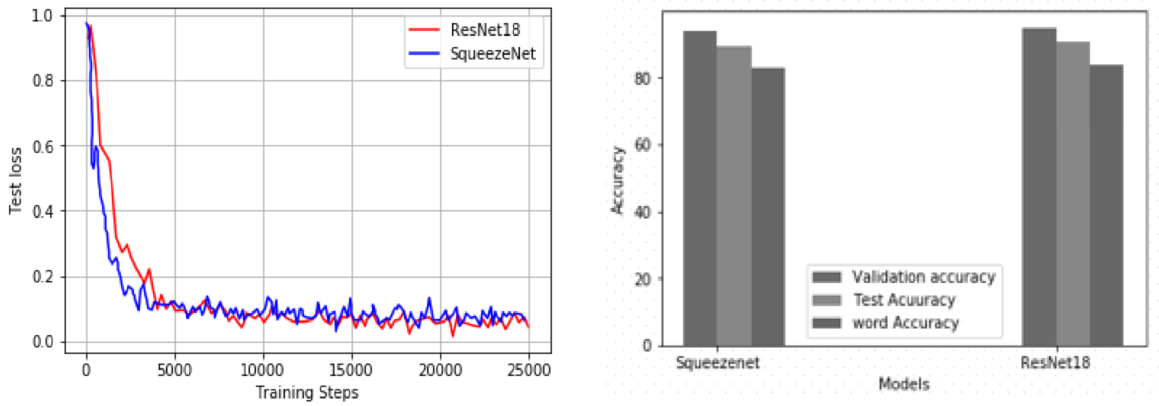
To compute the word recognition rates, we utilized 500 test pages authored by various individuals, containing nearly 1000 Telugu words. After detecting pages and words, these words underwent segmentation, followed by character recognition using the SqueezeNet and ResNet18 models. The achieved word recognition rates are detailed in Table 1.

Comparison of SqueezeNet and ResNet18 is shown in Figure 6. Figure 6(a) gives the loss plot comparison of ResNet18 and SqueezeNet. Figure 6(b) shows the validation, test, and word accuracy rate comparison of both the models. ResNet18 has slightly enhanced performance than SqueezeNet. But in terms of training time SqueezeNet requires less training time and trainable parameters compared to ResNet18.

Table 2 displays a performance comparison between our model and existing models [13], [14], [25]. Developing an OCR for Telugu, a complex language with numerous compound characters, poses a significant challenge, leading to limited research in this area. While the VGG model achieved notable recognition rates, its deep architecture requires a substantial number of trainable parameters. In contrast, our lightweight models, with fewer parameters, exhibit high recognition rates and perform efficiently with a reduced parameter count.

Table 1. Comparison of ResNet18 and SqueezeNet

Model	Parameters	Validation character accuracy	Test character accuracy	Word accuracy
ResNet 18	13,68,640	95%	90.580	84%
SqueezeNet	8,61,344	94%	89.438	83%



(a) (b)

Figure 6. Comparison of ResNet18 and Squeeze Net (a) loss rate (b) accuracy

Table 2. Comparison of ResNet18 and SqueezeNet

Name of the author	Type of text	Technique	Accuracy
Panyam Narahari Sastry	Handwritten Telugu characters	Nearest neighbour classifier	78%
Sarika.N	Handwritten Telugu Characters	VGG-16	92%
Minesh Mathew	Scene Telugu Text detection	Hybrid CNN-RNN (character-level)	86.2%
		Hybrid CNN-RNN (word-level)	57.2%
Proposed	Handwritten Telugu Text	ResNet 18(Character-level)	95%
		(word-level)	84%
		SqueezeNet (character-level)	94%
		(word-level)	83%

4. CONCLUSION

Identifying handwritten Telugu text poses significant challenges due to the variety of writing styles, overlapping characters, and the need to differentiate between similar structures within a large character set. The segmentation algorithm proposed in this study has demonstrated notable success in accurately separating overlapping characters, thereby enhancing word recognition rates. Among the models evaluated, ResNet18 and SqueezeNet have emerged as particularly effective for this task. ResNet18, with its deep architecture, achieves a character recognition rate (CRR) of 95% and a word recognition rate (WRR) of 84%. In contrast, SqueezeNet, despite having fewer parameters, achieves a CRR of 94% and a WRR of 83%. Both models capture the intricate features of Telugu text effectively. Moreover, their shorter training times compared to deeper models like VGG make them highly suitable for the digitalization of Telugu text.

REFERENCES

[1] J. Memon, M. Sami, R. A. Khan, and M. Uddin, "Handwritten optical character recognition (OCR) a comprehensive systematic literature review (SLR)," in *IEEE Access*, vol. 8, pp. 142642-142668, 2020, doi: 10.1109/ACCESS.

[2] M. Kumar, S. R. Jindal, M. K. Jindal, and G. S. Lehal, "Improved recognition results of medieval handwritten Gurmukhi manuscripts using boosting and bagging methodologies," *Neural Processing Letters*, vol. 50, pp. 43-56, Sep. 2018, doi: 10.1007/s11063-018-9913-6.

[3] C. Wolf, J. M. Jolion, and F. Chassaing, "Text localization, enhancement and binarization in multimedia documents," *2002 International Conference on Pattern Recognition*, vol. 2, pp. 1037-1040, 2002, doi: 10.1109/ICPR.2002.1048482.

[4] P. Sahare and S. B. Dhok, "Multilingual character segmentation and recognition schemes for Indian document images," in *IEEE Access*, vol. 6, pp. 10603-10617, 2018, doi: 10.1109/ACCESS.2018.2795104.

[5] U. Bhattacharya and B. B. Chaudhuri, "Handwritten numeral databases of indian scripts and multistage recognition of mixed numerals," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 444-457, March 2009, doi: 10.1109/TPAMI.2008.88.

[6] C. Naveena and V. N. M. Aradhya, "Handwritten character segmentation for Kannada scripts," *2012 World Congress on Information and Communication Technologies*, pp. 144-149, 2012, doi: 10.1109/WICT.2012.6409065.

[7] T. R. Babu, N. V. Rao, L. P. Reddy, T. S. K. Prabhu, and B. R. Babu, "Syllable segmentation of Telugu document images," *2010 2nd International Conference on Signal Processing Systems*, 2010, pp. 78-82, doi: 10.1109/ICSPS.2010.5555217.

[8] K. L. Banumathi and A. P. J. Chandra, "Line and word segmentation of Kannada handwritten text documents using projection profile technique," *2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT)*, Mysuru, India, 2016, pp. 196-201, doi: 10.1109/ICEECCOT.2016.7955214.

[9] N. S. Rani, A. C. Subramani, A. Kumar, and B. R. Pushpa, “Deep learning network architecture based Kannada handwritten character recognition,” *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 213-220, 2020, doi: 10.1109/ICIRCA48905.2020.9183160.
[10] R. S. Kunte and R. D. S. Samuel, “A simple and efficient optical character recognition system for basic symbols in printed Kannada text,” *Sadhana*, vol. 32, pp. 521–533, 2007, doi: 10.1007/s12046-007-0039-1.
[11] 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,” *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pp. 1494-1500, 2021, doi: 10.1109/ICESC51422.2021.9532658.
[12] N. Prameela, P. Anjusha, and R. Karthik, “Off-line Telugu handwritten characters recognition using optical character recognition,” *2017 International Conference of Electronics, Communication and Aerospace Technology (ICECA)*, pp. 223-226, 2017, doi: 10.1109/ICECA.2017.8212801.
[13] P. N. Sastry, T. R. V. Lakshmi, N. V. K. Rao, T. V. Rajinikanth, and A. Wahab, “Telugu handwritten character recognition using zoning features,” *International Conference on IT Convergence and Security (ICITCS)*, pp. 1-4, 2014, doi: 10.1109/ICITCS.2014.7021817.
[14] N. Sarika, N. Sirisala, and M. S. Velpuru, “CNN based optical character recognition and applications,” *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, pp. 666-672, 2021, doi: 10.1109/ICICT50816.2021.9358735.
[15] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, “Design of an image edge detection filter using the Sobel operator,” in *IEEE Journal of Solid-State Circuits*, vol. 23, no. 2, pp. 358-367, April 1988, doi: 10.1109/4.996.
[16] A. Choudhary, R. Rishi, and S. Ahlawat, “A new character segmentation approach for off-line cursive handwritten words,” *Procedia Computer Science*, vol. 17, pp. 88–95, 2013, doi: 10.1016/j.procs.2013.05.013.
[17] R. G. Casey and E. Lecolinet, “A survey of methods and strategies in character segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 690–706, Jul. 1996, doi: 10.1109/34.506792.
[18] Y. H. Liu, “Feature extraction and image recognition with convolutional neural networks,” *Journal of Physics: Conference Series, 1087, 062032,* doi: 10.1088/1742-6596/1087/6/062032.
[19] H. R. Mamatha and K. Srikantamurthy, “Morphological operations and projection profiles based segmentation of handwritten kannada document,” *International Journal of Applied Information Systems, IJAIS Journal*, vol. 4, no. 5, 2012, doi: 10.5120/ijais12-450704.
[20] M. Jogin, Mohana, M. S. Madhulika, G. D. Divya, R. K. Meghana, and S. Apoorva, “Feature extraction using convolution neural networks (CNN) and deep learning,” *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology (RTEICT)*, pp. 2319-2323, 2018, doi: 10.1109/RTEICT42901.2018.9012507.
[21] N. S. Rani, S. K. Verma, and A. Joseph, “A zone based approach for classification and recognition of Telugu handwritten characters,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 6, no. 4, pp. 1647-1653, 2016, doi: 10.11591/ijece.v6i4.10553.
[22] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778, 2016, doi: 10.1109/CVPR.2016.90.
[23] F. Iandola, M. Moskewicz, K. Ashraf, S. Han, W. Dally, and K. Keutzer, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and textless1MB model size,” *arXiv:1602.07360,* 2016.
[24] M. S. Velpuru, G. Tejasree, and M. R. Kumar, “Telugu handwritten character dataset,” *IEEE Dataport*, 2020, doi: 10.21227/Mw6a-D662.
[25] M. Mathew, M. Jain, and C. V. Jawahar, “Benchmarking scene text recognition in devanagari, Telugu and Malayalam,” *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 9, November 2017, pp. 9-15, doi: 10.1109/ICDAR.2017.364.

BIOGRAPHIES OF AUTHORS

Buddaraju Revathi currently working as an Assistant Professor at S.R.K.R Engineering college, holds an M.Tech Degree from S R K R Engineering College. Her academic pursuits center around wireless communication, natural language processing, and data science. She can be contacted at email: buddaraju.revathi@gmail.com.

B. N. V. Narasimha Raju currently working as an Assistant Professor at S.R.K.R Engineering college, holds an M.Tech Degree from S R K R Engineering College. Her academic pursuits center around natural language processing, machine translation, artificial intelligence and deep learning. He can be contacted at email: buddaraju.narasimharaju@gmail.com.

Boddu L V Siva Rama Krishna is working at SRM University-AP, Since 2023 September as Assistant Professor in the Department of Computer Science and Engineering. He is worked at Sagi Rama Krishnam Raju Engineering College from 2016 March 1st to 2023 August as an Assistant Professor in the Department of Computer Science and Engineering. Research area is data mining, machine learning, deep learning translation. he is a life member of computer society of India (CSI), and institute of engineers (IE). He can be contacted at email: sivaramakrishna.b@srmap.edu.in and krishna2928@gmail.com.

Ajay Dilip Kumar Marapatla is an Assistant professor in the Department of Computer Science and Engineering at SRM University. His research areas are Internet of Things, and Machine Learning. He has 12 Years of teaching experience. Currently he is pursuing Ph.D. from Pondicherry Central University, Pondicherry. He can be contacted at email: dileepmarapatla@gmail.com.

S. Suryanarayanaraju is presently working as an assistant professor in the Department of Information Technology, SRKR Engineering College (A), chinaamiram, Bhimavaram, Andhra Pradesh, India. He completed his B. Tech in 2008, M. Tech in 2010. He has published 7 journal papers and 2 conference papers. His research interests include datamining, artificial intelligence, machine learning, bigdata, and internet of things. He can be contacted at email: suryanarayanaraju.s@gmail.com.