Bulletin of Electrical Engineering and Informatics
Vol. 13, No. 6, December 2024, pp. 4115~4123
ISSN: 2302-9285, DOI: 10.11591/eei.v13i6.8170

Computationally efficient ResNet based Telugu handwritten text detection

Buddaraju Revathi[1,3], M. V. D. Prasad[1], Naveen Kishore Gattim[2]
[1]Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Guntur, India
[2]Department of Electronics and Communication Engineering, Sasi Institute of Technology and Engineering, Tadepalligudem, India
[3]Department of Electronics and Communication Engineering, S R K R Engineering College, Bhimavaram, India

Article Info

Article history:

Received Jan 11, 2024
Revised Apr 18, 2024
Accepted May 16, 2024

Keywords:

Bilateral filter
Convolution neural network
Deep learning
Inception
ResNet

ABSTRACT

Optical character recognition (OCR) is a technological process that converts diverse document formats into editable and searchable data. Recognition of Telugu characters through OCR poses a challenge because of compound characters. Identifying handwritten Telugu text proves difficult due to the substantial number of characters, their similarities, and overlapping forms. To handle overlapping characters, we implemented a segmentation algorithm that efficiently separates these characters, consequently enhancing the model's accuracy. Feature extraction is a crucial phase in recognizing a broader range of characters, especially those that are similar in appearance. So, we have employed a light weighted ResNet 34 model that effectively addresses these challenges and handles deep networks without declining accuracy as the network's depth increases. We have achieved a word level recognition rate of 81.5%. In addition, the parameters required by the model are less when compared to its counterpart inception V1, making it computationally efficient.

Corresponding Author:

Buddaraju Revathi
Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation
Vaddeswaram, Guntur, Andhra Pradesh-522502, India
Email: buddaraju.revathi@gmail.com

1. INTRODUCTION

Optical character recognition (OCR) involves transforming text located within images into a machine-readable format, enabling editing and processing by machines. As technology advances, there arises a demand for digitalization through the recognition of text from diverse sources like handwritten notes, forms, and printed papers, all presented as images, and converting them into an editable text format. Despite the advancements in OCR technology, the Telugu language, a significant Indian language, has not received adequate attention in research, resulting in insufficient solutions for recognizing its handwritten script. This paper aims to address this shortfall by employing advanced deep learning (DL) models, specifically ResNet architecture known for their effectiveness in image recognition tasks.

OCR involves progressing through five specific stages: preprocessing, word detection, character segmentation, feature extraction, and recognition [1], [2]. Preprocessing improves image quality which is achieved through noise reduction, contrast enhancement, and skew correction [3], [4]. Recognizing text regions involves isolating them from non-text areas, effectively reducing the processing load. Segmenting text regions into individual characters ensures independent processing, eliminating ambiguity and confusion [5]. High accuracy in this step is vital for the overall success of OCR. In character recognition, convolution neural networks (CNNs) are vital due to their ability to independently acquire relevant features, navigate spatial hierarchies, ensure translation invariance, and effectively handle the intricate details within character images [6], [7]. Their expertise in learning from data establishes them as indispensable elements of modern character recognition systems. Training OCR systems using labeled datasets enhances recognition capabilities, making the model more adaptable to various fonts, styles, and languages. This process significantly improves OCR performance, ensuring superior results.

Researchers have focused their efforts on developing OCR systems customized for indigenous languages. A significant challenge they encounter, particularly with limited datasets, is accurately segmenting characters. Additionally, identifying a suitable model adept at extracting intricate features unique to the native language poses a major hurdle in the development process. Segmentation of overlapping characters enhances the recognition rates of OCR. Garain and Chaudhuri [8] has developed a predictive algorithm to efficiently choose potential cut columns for separating interconnected characters of Devnagari and Bangla scripts. This approach relies on fuzzy multifactorial analysis. Song *et al*. [9] developed a method to segment merged characters. This technique leverages data extracted from the front part of combined characters to anticipate potential options for the left character. It then employs adaptive masking followed by recognition to confirm this prediction on character. Consequently, the critical path, tailored to arbitrary shapes, aligns with the right contour of the leftmost character, ensuring the preservation of the next character's shape.

To address the second issue, the advancement of DL techniques has drawn the attention of researchers to integrate them into OCR systems. Neural networks are powerful feature extractors. Deep neural networks are capable of extracting features of orthographic structured characters. Malhotra and Addis [10] used a DL technique to identify handwritten Ethiopic texts. This model embraces an end-to-end methodology, allowing seamless characteristic extraction and effective detection in subsequent steps. The core of the model comprises an attention technique combined with a connectionist temporal classification. Furthermore, the model incorporates seven CNNs and two recurrent neural networks (RNNs). Rasheed *et al.* [11] introduced a technique for the detection of Urdu characters and digits which are handwritten. It enhanced precision by applying the principles of transfer learning using pre-trained CNNs. Chandio *et al*. [12] suggested a model that takes an entire word image as input, bypassing the need for prior segmentation into single characters, and converted them to sequence of pertinent features. This method comprises of a deep CNN and a RNN to capture structural features. Connectionist temporal classification method is employed to align the forecasted sequences with the desired target labels. Zhang *et al.* [13] used a sophisticated CNN model for identification of text, utilizing a sliding window approach. Subsequently, a text recovery process detects likenesses within the image and establishes the location of the identified text within the database. Mathew *et al.* [14] proposed an OCR technique designed to recognize cropped images containing words in Telugu. This method employs a hybrid CNN-RNN architecture for word transcription, where convolutional layers initially generate feature vectors in a column-wise manner. The output from these convolutional layers then undergoes recurrent layer processing, utilizing deep bidirectional long short-term memory (BLSTM) networks for making predictions. Subsequently, the predictions are passed through transcription layers, generating label sequences based on the RNN layer predictions. With the utilization of the hybrid CNN-RNN model achieves a WRR of 57.2% and a character recognition rate (CRR) of 86.2%.

The segmentation algorithm employed to separate overlapping characters must take into account the linguistic intricacies particular to the language. Hence, we have devised a character segmentation algorithm customized for Telugu. This is done by considering the limited existing work in this language and the differences in linguistic characteristics. Earlier researchers employed deep models with extensive training durations for native languages [6], [15]. In response, we have developed a lightweight model is finely tuned for accurately discerning Telugu language characteristics. We utilize a light weighted ResNet neural network that extracts the features of the intricate Telugu characters without overfitiing. Moreover, we employ the dictionary method to resolve the problem of improper character segmentation by rectifying errors in words.

2. RESNET BASED CHARACTER RECOGNITION

2.1. Page and word detection

In the first stage, the image undergoes grayscale conversion, after which the system identifies page borders to rectify any skew detected in the image. The bilateral filter is designed to realm edges while decreasing noise, making it effective for smoothing images [16]. Mathematically it is given by (1):

$$F(t) = \frac{1}{W_s} \sum_{u \epsilon \Omega} G_{\sigma_m}(\|t-u\|) \times G_{\sigma_n}(\|I(t)-I(u)\|) \times I(u) \tag{1}$$

Where are pixel intensities at *t*, *u* of image, $I(t)$, $I(u)$, Ω represents spatial neighbourhood at *t* pixel, G_{σ_m}, G_{σ_n} are standard deviations of σ_m, σ_n of spatial domain and intensity domain. W_s is normalizarion factor.

Gaussian thresholding technique identifies page edges by considering variations in illumination. It utilizes a threshold function that reflects changes in illumination, enabling the accurate detection of page boundaries. The Median blur removes the unnecessary details. Consider an input image *I(y,z)* and a kernel of size R×R, the median blur operation at a specific pixel *(y,z)* can be expressed as in (2):

$$O(y,z) = M(\{I(y',z') \mid y' \in [t - \lfloor R/2 \rfloor, y + \lfloor R/2 \rfloor],\ z' \in [z - \lfloor R/2 \rfloor, z + \lfloor R/2 \rfloor]\}) \qquad (2)$$

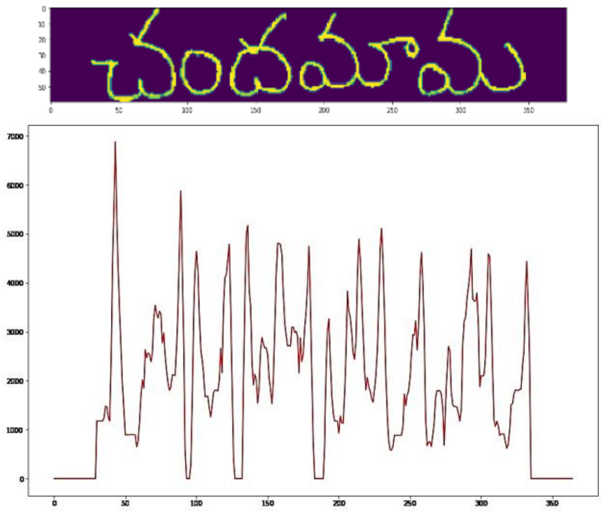
Where $O(y,z)$ signifies the pixel intensity obtained as the result of applying the median blur operation at the position *(y, z)*, *⌊R/2⌋* represents *R* by 2 division, defining the extent of the kernel's boundary around the pixel *(y, z)*. *M({…})* computes the median by selecting the middle value from the set of pixel intensities within the kernel.

Canny edge detection reveals the structural details of inherent formation within the image [17]. Morphological operations are then employed to extract the image boundaries by identifying variations in grey levels. Subsequently, contours are outlined, distinguishing the page from its background. To identify words on the page, eliminate page noise using the Sobel operator, which relies on a 5×5 Gaussian filter for precise filtering. The union and intersection methods are employed to form bounding boxes around words. The count of these bounding boxes determines the total number of words on the specific page.

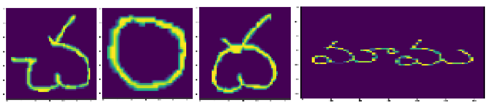
2.2. Character segmentation and feature extaraction

Character segmentation is an important step in OCR [18], [19]. The primary challenge associated with handwritten text lies in the occurrence of overlapping characters within a word. When employing vertical projections, effectively segmenting overlapping characters becomes impractical. To address this issue, we have devised a segmentation algorithm specifically tailored to accommodate the unique linguistic features of the Telugu language.

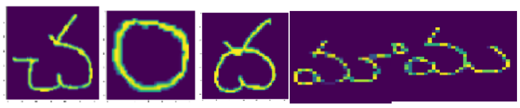
Figure 1 shows character segmenation graphs. Figure 1(a) shows the input word, and it is vertical projection graph. Figure 1(b) shows the output with vertical projection profile and Figure 1(c) shows the output of our character segmentation algorithm. The developed alogirithm can preserve the most important features of the base character.



(a)



(b)



(c)

Figure 1. Character segmentation: (a) vertical projection profile graph for a word, (b) segments obtained based on vertical projection graph, and (c) segments obtained after passing through our algorithm

An algorithm for segmenting words to characters is explained.

Step 1: let I_{width} be the pixel intensities calculated along the width. Compute the cumulative sum S_i at each position *i* along the width using (3):

$$S_i = \sum_{j=0}^{i} I_{width}(j) \quad (3)$$

Store the cumulative sums in a list $L_{cumulative}$.
Step 2: split the list $L_{cumulative}$e into sub lists whenever the value is zero. Let $L_{sublists}$ represent the list of sub lists obtained.
Step 3: for each sub list L_i in $L_{sublists}$ calculate the character widths W_i using (4):

$$W_i = L_i(k) - L_i(k-1) \quad (4)$$

Where *k* is the index of sub list L_i.
Step 4: find the minimum character width W_{min}among all W_i. Set a threshold 2× W_{min}. Compare each character width W_i with the threshold.
Step 5: if threshold=2× W_{min}. Calculate the mid width of image. Find sum of pixel intensities 10 values to right and left starting from middle point. Segment the image at minimum sum pixel intensity.

A CNN is a specialized neural network that was meticulously developed to process organised grid data [20]. CNN's demonstrate exceptional proficiency in grasping complex spatial features and hierarchies within the input data. In CNN, the convolution operation consists of applying a filter to an input image. Mathematically, this operation can be represented as in (5):

$$C(c,d) = \sum_{r=1}^{R} \sum_{s=1}^{S} IM(c+r, d+s) \times K(r,s) \quad (5)$$

Where *IM* signifies the input image, *K* represents the filter, and *C* denotes the feature map generated at the output by moving the filter across the input image, conducting element-wise multiplications, and then summing the results. Following the convolution step, a non-linear activation function is utilized to introduce non-linearity. Rectified linear unit (ReLU) is mostly used in CNNS and is given by (6):

$$ReLU(z) = max(0, z) \quad (6)$$

Max pooling layers decrease spatial dimensions. Specifically, max pooling selects the highest value from a cluster of pixels, represented mathematically as in (7):

$$MaxPooling(t,u) = max(T\ t,u)\ for\ t \in [1,T], u \in [1,U] \quad (7)$$

Fully connected layers execute advanced reasoning processes. The output is computed utilizing weights and biases given by (8):

$$M = W \cdot N + c \quad (8)$$

where *M* is the output and *N* signifies the input and *W* is the weights and *c* show added bias. The SoftMax function computes class probabilities based on raw scores given by (9):

$$P_x = \frac{e^{Z_x}}{\sum_{y=1}^{C} e^{Z_y}} \quad (9)$$

where P_x represents class probabilities, Z_x are raw scores, and *C* is number of classes.

Historically, many researchers favored SVM classifiers and diverse CNN architectures for native language OCR tasks. However, SVM's recognition rate was relatively modest, and CNN models demanded extensive training durations due to their depth. To address these issues, we opted for ResNet models, adjusting filters to suit our needs. The selection of filters was meticulously guided by examining accuracy and loss metrics across various layers. Our proposed model, depicted in Figure 2, utilizes fewer parameters while achieving accuracy rates comparable to existing models [6].

We have used ResNet architecture for extraction of features. ResNet 34 consists of 32 intermediate convolutional layers and 2 Pooling layers. ResNet-34 leverages the strength of deep residual connections, enabling it to adeptly handle the challenges associated with training deep networks [21]. The shortcut connections embedded within its residual blocks ensure a seamless flow of gradients, successfully addressing the vanishing gradient problem. Additionally, ResNet-34 employs a mix of block types, such as Basic and Bottleneck structures, carefully balancing the complexity of the model with computational efficiency. These features of the ResNet makes them capable of extracting the features of native languages effectively, especially Telugu language. The residual module is shown in Figure 3. In the given diagram, the variable *z* symbolizes the block's input. This input traverses the convolutional path, denoted as *G(z)*, and the outcome is summed with the original input z. This fusion produces the block's output, represented as *R(z)*.

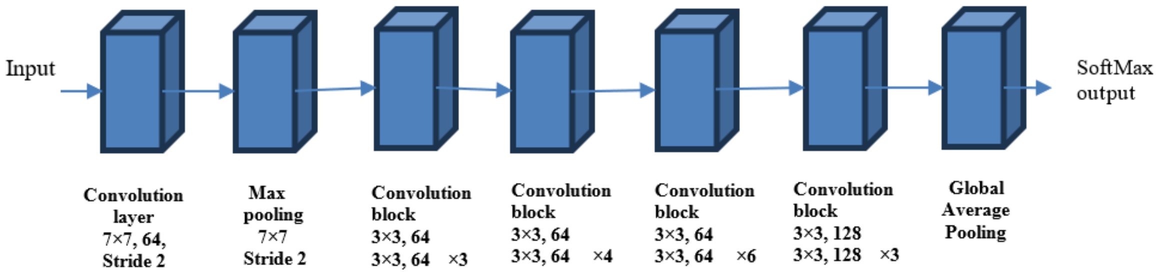


Figure 2. Architecture of ResNet34

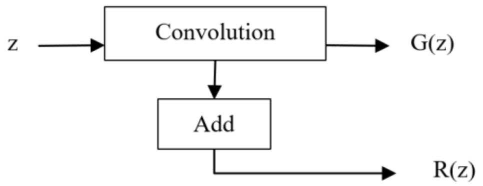


Figure 3. Residual module

2.3. Dictionary model

In Telugu, certain characters, like య and మ, స and న, and ఢ and ధ, introduce confusion due to their visual resemblance. The are some examples of confusion characters. The diversity in individuals' handwriting styles adds complexity to precise character prediction, resulting in reduced recognition rates. To overcome this obstacle, we have created specialized dictionary models tailored to handle challenges associated with ambiguous characters. As a result, these models play a crucial role in enhancing word recognition rates (WRR). The dictionary model algorithm is outlined as follows.

Let C Set of candidate characters for a given position from the OCR engine. D is Dictionary of valid words. $S(c)$ is scoring function that assigns a score to candidate character c.

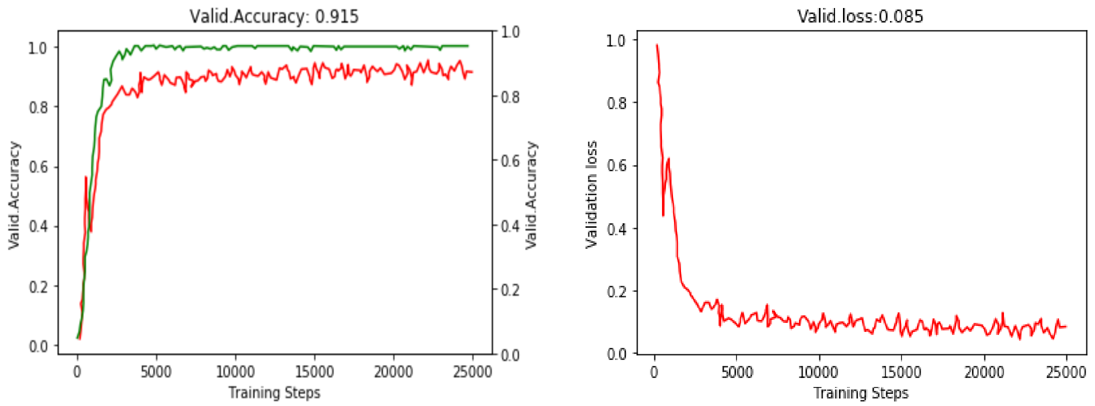
1. Initialize best candidate to null.
2. For each candidate character c in C do:
    - If $c \in D$ then:
    - Calculate the updated score $S'(c) = S(c) + \Delta$, where Δ is increment factor.
    - If Best Candidate is null or $S'(c)$ is greater than the score of best candidates, then:
        - Set Best Candidate to c.
    - Else (if c is not in D) then:
        - Continue to the next iteration.
3. Output Best Candidate as the recognized character for the given position.

3. RESULTS

Languages spoken regionally often incorporate a wider variety of characters and words compared to English. Consequently, even with a smaller dataset, developing an OCR system becomes viable through character-level training. However, the challenge arises in the character segmentation process, a critical step that must effectively separate overlapping characters. If not executed accurately, the OCR system's performance significantly deteriorates. The precision of character segmentation plays a pivotal role in ensuring the OCR system can faithfully recognize and interpret characters in regional languages, which are inherently complex due to their diverse characters and intricate script structures. The proposed segmentation method aims to successfully separate overlapped Telugu characters within a word.

We have compared the performance of the ResNet 34 with inception V1 models. Inception V1 extracts the features of image at multiple scales [22]. The dataset is from IEEE Dataport [23]. The samples within the dataset are resized to a uniform size of 64×64 pixels. 11,602 samples are used as training samples 2,565 samples are used for validation set. Both the models are trained for 25,000 training steps.

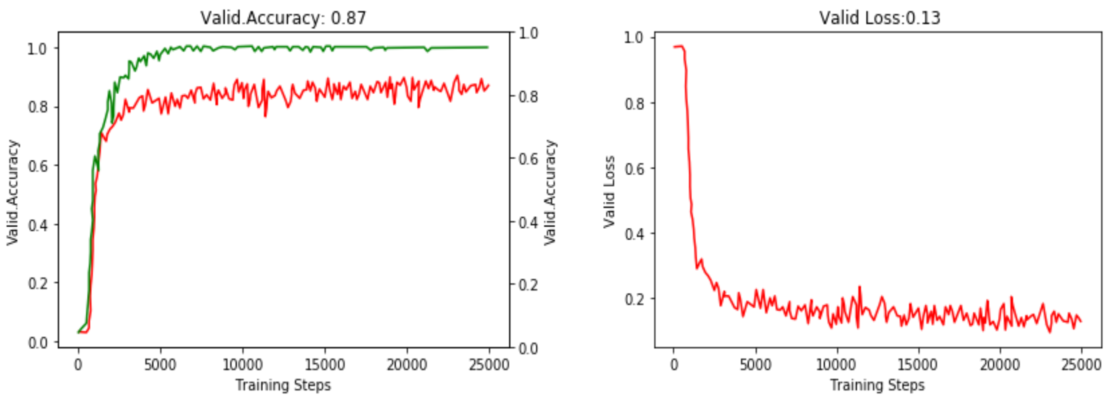
Figure 4 shows plots of ResNet 34. The accuracy plot of Resnet 34, displayed in Figure 4(a), exhibits a remarkable accuracy rate of 91.5% after 25,000 training steps. The loss plot of Resnet 34, depicted in Figure 4(b), illustrates a validation loss of 8.5%.



(a) (b)

Figure 4. Plots of ResNet 34; (a) recognition accuracy of ResNet 34 and (b) loss plot of ResNet34

Figure 5 shows plots of inception V1. The accuracy graph for inception V1, as depicted in Figure 5(a), demonstrates an impressive accuracy of 87% following 25,000 training steps. Similarly, the loss graph in Figure 5(b) reveals a validation loss of 13%. The plot demonstrates that ResNet 34 achieves a higher accuracy rate than inception V1.



(a) (b)

Figure 5. Plots of inception V1; (a) recognition accuracy of inception V1 and (b) loss plot of inception V1

We have also used a test dataset to evaluate inception V1 and ResNet 34 performance. Inception V1 has test accuracy of 80.5% and ResNet 34 has test accuracy of 87.94%. To know the word level accuracy rates the models are tested for 1,000 words in which ResNet 34 attained 80%, while inception V1 achieved 76% without using dictionary models. We have used dictionary model to avoid accuracy rates getting effected by the confusion characters. With dictionary model, ResNet 34 has achieved 81.5% and inception V1 attained 78%-word accuracy rates. This comparison of various parameters is shown in Table 1.

Figure 6 shows comparison of ResNet 34 and inception V1. Figure 6(a) presents a comparison of validation loss between inception V1 and ResNet 34. The plot distinctly indicates that ResNet 34 exhibits lower loss and superior performance in comparison to inception V1. In Figure 6(b), the character and word accuracy rates of ResNet 34 and inception V1 are displayed. It is evident from the figure that ResNet 34 exhibits superior character and WRRs compared to inception V1.

Table 1. Performance comparison of ResNet 34 and inception V1

Model	Parameters	Validation character accuracy (%)	Test character accuracy	Word accuracy (%)	Word accuracy with dictionary (%)
ResNet 34	13,32,480	91.5	87.94	80	81.5
InceptionV1	56,31,016	87	80.5	76	78

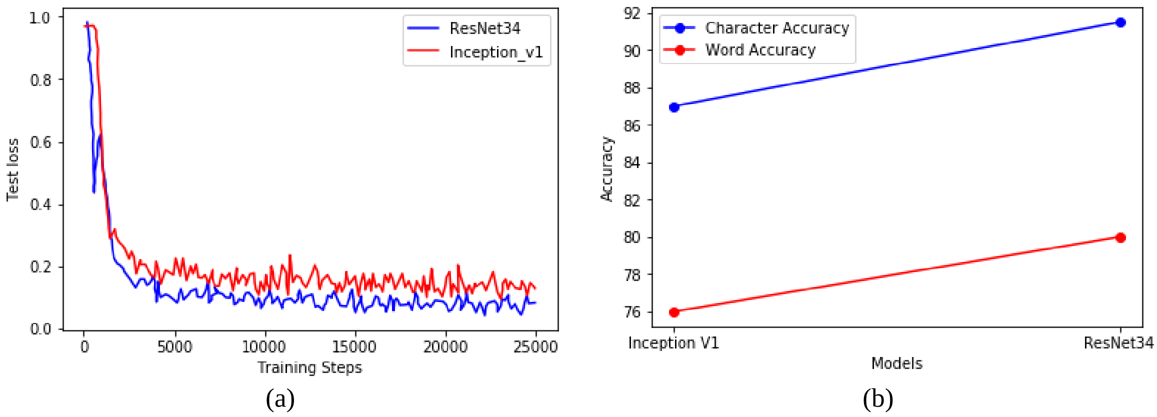


Figure 6. Comparison of ResNet 34 and inception V1; (a) comparison of validation loss plots and (b) recognition rates of ResNet 34 and inception V1

Figure 7 shows the plots of inception V1 vs ResNet 34. Figure 7(a) illustrates a comparison of word accuracy with and without dictionary between ResNet 34 and inception V1. It is evident from the plot that ResNet 34 outperforms inception V1 in terms of accuracy. In Figure 7(b), the parameter comparison between ResNet 34 and inception V1 is presented. ResNet 34 is equipped with 1,332,480 parameters, whereas inception V1 comprises 56,31,016 parameters. So, it is clear that ResNet 34 attained a significant accuracy rate with fewer parameters than inception V1. ResNet 34 is computationally efficient than inception V1 by recognizing more Telugu characters with less training time.

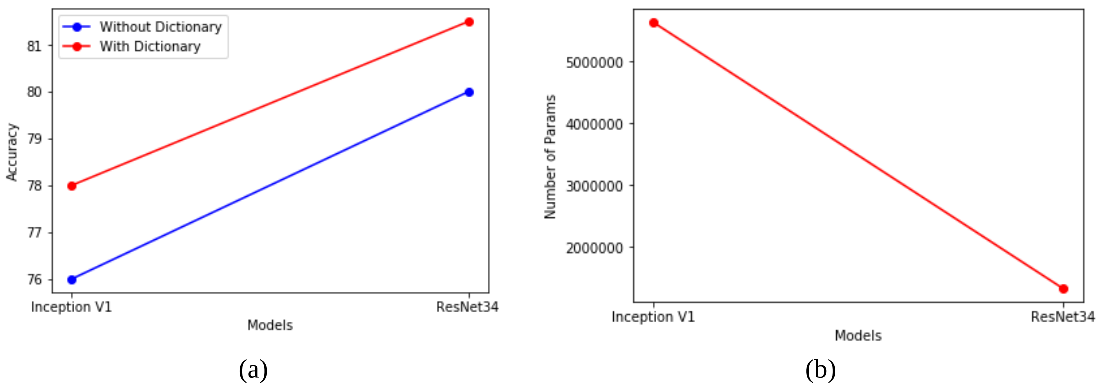


Figure 7. Inception V1 vs ResNet 34; (a) recognition rates with and without dictionary and (b) parameters of ResNet 34 and inception V1

Previous studies did not yield a noteworthy recognition rate for handwritten text. Table 2 illustrates the performance of different techniques on Telugu languages [14], [24], [25]. It is evident from Table 2 that ResNet surpasses other models in both CRR and WRR.

Table 2. Performance of various techniques on Telugu OCR

Name of the author	Type of text	Technique	Accuracy (%)
Rao and Negi [24]	Handwritten Telugu text	Hidden Markov models (HMMs), Akshara models and Akshara Bigram language models	74
Rani *et al.* [25]	Handwritten Telugu characters	SVM classifier	80
Mathew *et al.* [14]	Scene Telugu	Hybrid CNN-RNN (CRR)	86.2
	Text detection	Hybrid CNN-RNN (WRR)	57.2
Proposed	Handwritten	CNN, ResNet 34 (CRR)	91.5
	Telugu text	CNN, ResNet 34 (WRR)	81.5

4. CONCLUSION

OCR for regional languages often faces challenges stemming from limited resources, imbalanced datasets, and character resemblances. ResNet34 emerges as a powerful feature extractor adept at discerning subtle character differences and handling imbalanced data sets. Leveraging effective preprocessing techniques to reduce noise, correct page skew, and enhance input quality, alongside a character segmentation algorithm, contributes to improved WRR by precisely breaking down words into individual characters.

In the realm of handwritten Telugu text, ResNet34 has exhibited a remarkable efficiency of 91.5% in character recognition and 80% in word recognition by extracting features from orthographic characters. ResNet 34 requires 13,32,480 trainable parameters compared to inception V1 that consumes 56,31,016 parameters. Notably, ResNet34 showcases exceptional computational efficiency when compared to its counterpart, inception V1. Furthermore, the integration of a dictionary model played a crucial role in correcting characters, particularly in cases of similarity, thereby elevating the overall accuracy of the OCR system to 81.5%. Considering the training time, accuracy rates, and model performance on the test set, ResNet architectures are well-equipped to proficiently recognize handwritten Telugu text, provided appropriate model parameters are selected based on dataset size and class distribution.

REFERENCES

[1] M. Mathew, A. K. Singh, and C. V. Jawahar, “Multilingual OCR for indic scripts,” in *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, IEEE, Apr. 2016, pp. 186–191, doi: 10.1109/DAS.2016.68.

[2] M. Kumar, M. K. Jindal, and R. K. Sharma, “Review on OCR for handwritten indian scripts character recognition,” in *Communications in Computer and Information Science*, 2011, pp. 268–276, doi: 10.1007/978-3-642-24055-3_28.

[3] A. K. Das and B. Chanda, “A fast algorithm for skew detection of document images using morphology,” *International Journal on Document Analysis and Recognition*, vol. 4, no. 2, pp. 109–114, Dec. 2001, doi: 10.1007/PL00010902.

[4] B. B. Chaudhuri and U. Pal, “Skew angle detection of digitized Indian script documents,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 182–186, 1997, doi: 10.1109/34.574803.

[5] J. Ryu, H. Il Koo, and N. I. Cho, “Word segmentation method for handwritten documents based on structured learning,” *IEEE Signal Processing Letters*, vol. 22, no. 8, pp. 1161–1165, Aug. 2015, doi: 10.1109/LSP.2015.2389852.

[6] N. Shaffi and F. Hajamohideen, “uTHCD: A new benchmarking for tamil handwritten OCR,” *IEEE Access*, vol. 9, pp. 101469–101493, 2021, doi: 10.1109/ACCESS.2021.3096823.

[7] Y. Weng and C. Xia, “A new deep learning-based handwritten character recognition system on mobile computing devices,” *Mobile Networks and Applications*, vol. 25, no. 2, pp. 402–411, Apr. 2020, doi: 10.1007/s11036-019-01243-5.

[8] U. Garain and B. B. Chaudhuri, “Segmentation of touching characters in printed devnagari and bangla scripts using fuzzy multifactorial analysis,” *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 32, no. 4, pp. 449–459, Nov. 2002, doi: 10.1109/TSMCC.2002.807272.

[9] J. Song, Z. Li, M. R. Lyu, and S. Cai, “Recognition of merged characters based on forepart prediction, necessity-sufficiency matching, and character-adaptive masking,” *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 1, pp. 2–11, Feb. 2005, doi: 10.1109/TSMCB.2004.837588.

[10] R. Malhotra and M. T. Addis, “End-to-end historical handwritten ethiopic text recognition using deep learning,” *IEEE Access*, vol. 11, pp. 99535–99545, 2023, doi: 10.1109/ACCESS.2023.3314334.

[11] A. Rasheed, N. Ali, B. Zafar, A. Shabbir, M. Sajid, and M. T. Mahmood, “Handwritten Urdu characters and digits recognition using transfer learning and augmentation with AlexNet,” *IEEE Access*, vol. 10, pp. 102629–102645, 2022, doi: 10.1109/ACCESS.2022.3208959.

[12] A. A. Chandio, M. Asikuzzaman, M. R. Pickering, and M. Leghari, “Cursive text recognition in natural scene images using deep convolutional recurrent neural network,” *IEEE Access*, vol. 10, pp. 10062–10078, 2022, doi: 10.1109/ACCESS.2022.3144844.

[13] D. Zhang, Y. Liu, Z. Wang, and D. Wang, “OCR with the deep CNN model for ligature script-based languages like Manchu,” *Scientific Programming*, pp. 1–9, Jun. 2021, doi: 10.1155/2021/5520338.

[14] M. Mathew, M. Jain, and C. V. Jawahar, “Benchmarking scene text recognition in Devanagari, Telugu and Malayalam,” in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, IEEE, Nov. 2017, pp. 42–46, doi: 10.1109/ICDAR.2017.364.

[15] S. Drobac and K. Lindén, “Optical character recognition with neural networks and post-correction with finite state methods,” *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 23, no. 4, pp. 279–295, Dec. 2020, doi: 10.1007/s10032-020-00359-9.

[16] M. Zhang and B. K. Gunturk, “Multiresolution bilateral filtering for image denoising,” *IEEE Transactions on Image Processing*, vol. 17, no. 12, pp. 2324–2333, Dec. 2008, doi: 10.1109/TIP.2008.2006658.

[17] X. Ma, B. Li, Y. Zhang, and M. Yan, “The canny edge detection and its improvement,” in *Lecture Notes in Computer Science*, 2012, pp. 50–58, doi: 10.1007/978-3-642-33478-8_7.

[18] 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.

[19] H. R. Mamatha and K. Srikantamurthy, "Morphological operations and projection profiles-based segmentation of handwritten kannada document," *International Journal of Applied Information Systems*, vol. 4, no. 5, pp. 13–19, Oct. 2012, doi: 10.5120/ijais12-450704.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017, doi: 10.1145/3065386.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[22] C. Szegedy *et al.*, "Going deeper with convolutions," *arXiv preprint arXiv: 1409.4842*, Sep. 2014.

[23] S. V. Muni, G. Tejasree, and K. M. Ravi, "Telugu handwritten character dataset," *IEEE Dataport*, 2020.

[24] D. K. Rao and A. Negi, "Orthographic properties based telugu text recognition using hidden markov models," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, IEEE, Nov. 2017, pp. 45–50, doi: 10.1109/ICDAR.2017.327.

[25] 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, Aug. 2016, doi: 10.11591/ijece.v6i4.10553.

BIOGRAPHIES OF AUTHORS

Buddaraju Revathi is a research scholar at Koneru Lakshmaiah Education Foundation and has received M.Tech. degree from S. R. K. R. Engineering College. She is presently working as an assistant professor at S.R.K.R Engineering college. Her areas of interest are wireless communication, natural language processing, and data science. He can be contacted at email: buddaraju.revathi@gmail.com.

M. V. D. Prasad is working as an associate professor within the signal and image processing research group at the Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation University. Possessing a Ph.D. in Electronics and Communication Engineering, he is a highly skilled education professional. His specializes in computer vision, image processing, natural language processing, speech technology, artificial intelligence, and data science. Leveraging extensive research experience, he has contributed to esteemed journals with his publications. He can be contacted at email: mvd_ece@kluniversity.in.

Naveen Kishore Gattim has completed his Ph.D. from KL University. He is a faculty member at Sasi Institute of Technology and Engineering. His areas of interest are artificial intelligence and deep learning, image processing and data science. He can be contacted at email: drnaveen@sasi.ac.in.