Comparative Analysis of Classification Algorithms for Printed script

Arey Vikyath Reddy
Information Technology
Vardhaman College of Engineering
(JNTU Hyderabad)
Hyderabad, India
vikyath9256@gmail.com

Pulimamidi Nikhitha
Information Technolgy
Vardhaman College of Engineering
(JNTU Hyderabad)
Hyderabad, India
pulimamidinikhitha2003@gmail.com

Kadam Rahul
Information Technology
Vardhaman College of Engineering
(JNTU Hyderabad)
Hyderabad, India
rnani0264@gmail.com

Muni Sekhar Velpuru
Information Technology
Vardhaman College of Engineering
(JNTU Hyderabad)
Hyderabad, India
munisek@gmail.com

***Abstract*—A robust system for translating Telugu text from images to English is developed using Optical Character Recognition (OCR) and deep learning techniques. Tesseract OCR is utilized for detecting Telugu script, followed by translation with Facebook's mBART, a multilingual sequence-to-sequence model. Google Translate is also used for comparison purposes. The translation model's performance is evaluated with the BLEU score, which measures the similarity between machine translations and human-generated references.The system was tested on a dataset of Telugu images with corresponding English meanings. Various preprocessing techniques, such as resizing, filtering, and text segmentation, were applied to improve OCR accuracy. The outcomes demonstrate the effectiveness of integrating traditional OCR methods with advanced neural machine translation, providing a scalable solution for Telugu-to-English translation across various applications like document digitization, education, and accessibility.**

Keywords—component, formatting, style, styling, insert (key words**)**

I. INTRODUCTION

Optical Character Recognition (OCR) has advanced significantly, particularly in multilingual text recognition. However, Gujarati remains challenging due to its intricate curves, loops, modifiers, and joint characters. Despite ongoing research, achieving high accuracy in Gujarati character recognition remains difficult.While Convolutional Neural Networks (CNNs) have been effective for many languages, their performance on Gujarati has been limited. This study introduces two specialized CNN models: CNN-PGC for printed Gujarati characters and CNN-HGC for handwritten Gujarati characters. These models show notable improvements, achieving recognition rates of 98.08% for printed characters and 95.24% for handwritten numerals. The findings suggest that these models outperform traditional approaches and existing benchmarks, significantly enhancing OCR accuracy for Gujarati script [1].

Handwritten character recognition, especially in offline settings, presents unique challenges due to variations in handwriting influenced by factors such as individual writing styles, mood, and other external elements. Indian languages like Telugu and Hindi pose additional difficulties due to their cursive scripts and numerous diacritics.While significant progress has been made in recognizing languages like English and Chinese, Indian scripts still face challenges in achieving high accuracy. Recent studies have reported approximately 80% accuracy in recognizing handwritten Telugu and Hindi characters. However, there is still a need for further advancements to improve both recognition accuracy and processing efficiency [2].

Handwritten character recognition, especially in offline settings, presents unique challenges due to variations in handwriting influenced by factors such as individual writing styles, mood, and other external elements. Indian languages like Telugu and Hindi pose additional difficulties due to their cursive scripts and numerous diacritics. While significant progress has been made in recognizing languages like English and Chinese, Indian scripts still face challenges in achieving high accuracy. Recent studies have reported approximately 80% accuracy in recognizing handwritten Telugu and Hindi characters. However, there is still a need for further advancements to improve both recognition accuracy and processing efficiency [3].

For languages such as Urdu, OCR and word-spotting systems encounter additional difficulties due to script complexity, similar to Arabic and Persian. Clustering techniques for Urdu document images, unlike those for Latin scripts, have been less explored. A comprehensive study on clustering methodologies for Urdu revealed the critical role of segmentation in clustering-based indexing systems. Performance was evaluated using various metrics, including the CalinskiHarabasz, DavisBouldin, and Dunn indexes [4].

Recent advancements in deep learning and image processing have greatly improved Telugu character recognition, a challenging script due to its cursive structure and complex diacritics. A newly proposed model, DLTCRPHWC (Deep Learning Telugu Character Recognition for Printed and Handwritten Characters), integrates

ISBN: 978-93-343-1044-3

EfficientNet and CapsuleNet for feature extraction while utilizing a BiLSTM model for recognition. This approach has demonstrated superior performance compared to existing state-of-the-art methods, effectively recognizing both printed and handwritten Telugu characters within a single image [5].

II. Related works

Developing OCR systems for Indian scripts like Gujarati is challenging due to complex curves, modifiers, and joint characters. Traditional methods struggle with these intricacies, leading researchers to explore deep learning techniques such as CNNs. While CNNs have been applied to various languages, their effectiveness for Gujarati remains limited. This study introduces two optimized CNN models: CNN-PGC for printed Gujarati characters and CNN-HGC for handwritten numerals. These models achieve significant accuracy improvements, with CNN-PGC showing an 18.29% gain and CNN-HGC improving by 7.60% for handwritten numerals and 14.6% for handwritten characters.Handwritten character recognition remains difficult due to variations in writing styles. While CNNs and pre-trained models have improved recognition, Indian languages like Telugu and Hindi still face challenges due to their cursive nature and diacritics. Some studies report around 80% accuracy, but further enhancements are needed.The proposed CNN-based approach extracts textual features using convolutional layers and ReLU activation, followed by Dense layers with Softmax activation. Testing with different optimizers and classifiers has shown promising results. Expanding this method to more character classes and larger datasets could further improve OCR accuracy for Indian scripts
[1].

Offline handwritten character recognition faces challenges due to variations in writing styles, influenced by factors like mood and individual habits. While advancements in Convolutional Neural Networks (CNNs) and pre-trained models have improved recognition accuracy and reduced training time, research on Indian languages such as Telugu and Hindi has been slower due to their cursive scripts and complex diacritics. Although some studies have reached accuracy levels around 80%, there is still potential for improvement. This study introduces a CNN-based model that extracts textual features through convolutional layers, kernel filters, and ReLU activation, followed by Dense layers with Softmax activation. Evaluations using optimizers like SGD and Adam, along with classifiers like Random Forest and KNN, and a categorical crossentropy loss function, have shown promising results. Extending this approach to cover more character classes and larger datasets could further improve recognition accuracy [2].

This evaluation compares page analysis and recognition techniques for historical Bengali documents, focusing on OCR performance. It reports results for five methods, including submissions, a re run, and an open-source system. The analysis highlights the challenges of recognizing historical text and uses new character accuracy measures to assess OCR performance. While deep learning methods show promise, significant obstacles remain in processing older materials. The evaluation also reviews page segmentation and region classification, with general segmentation accuracy around 70%, led by the ABCD method. Google's OCR engine outperforms others with over 80% accuracy, though segmentation techniques for historical documents still lag behind contemporary materials [3]. The comparative analysis of page analysis and recognition techniques for historical Bengali documents focuses on OCR performance. It presents results from five methods, including three submissions, one re-run, and an open-source system, evaluated using various metrics and innovative character accuracy measures. Despite the promise of deep learning, challenges remain in processing older texts. The evaluation also examines page segmentation, with general segmentation achieving 70% accuracy, led by the ABCD method. Google's OCR engine outperformed others with over 80% accuracy, while segmentation techniques for historical documents continue to lag behind those for modern texts [4]. Character recognition, essential for both machine-printed and handwritten text, has advanced with deep learning and image processing techniques. Telugu Character Recognition (TCR) poses unique challenges within optical character recognition (OCR). This study introduces a deep learning-based TCR model, DLTCR-PHWC, capable of recognizing both printed and handwritten Telugu characters in the same image. The model begins with adaptive fuzzy filtering for pre-processing, followed by line and character segmentation. EfficientNet and CapsuleNet are used for feature extraction, while the Aquila optimizer and bi-directional LSTM model are applied for recognition. Experiments on a Telugu character dataset show that DLTCR-PHWC outperforms existing methods [5]. Languages worldwide use a variety of scripts, making script identification crucial in multilingual and multi-script contexts for selecting the appropriate character recognition and document analysis techniques. To address this challenge, several automatic script identification methods have been developed, falling into two main categories: structure-based techniques and visual appearance-based techniques. This survey examines these approaches, including methods for handling online data and video text. However, research in this area remains limited, highlighting the need for further exploration, particularly concerning handwritten documents
[6].

Languages around the world utilize diverse scripts, making the identification of these scripts essential in contexts involving multiple languages and scripts. This process is critical for choosing the right character recognition and document analysis methods. To tackle this issue, various automatic script identification methods have been developed, generally classified into two broad categories: structure-based and appearance-based techniques. This review explores these approaches, covering solutions for handling online data and text in videos. Despite the progress made, research in this field is still in its early stages, indicating a need for more in-depth studies, especially with regard to handwritten documents [7]. Languages around the world utilize diverse scripts, making the identification of these scripts essential in contexts involving multiple languages and scripts. This process is critical for choosing the right character recognition and document analysis methods. To tackle this issue, various automatic script identification methods have been developed, generally classified into two broad categories: structure-based and appearance-based techniques. This review explores these approaches, covering solutions for

ISBN: 978-93-343-1044-3

handling online data and text in videos. Despite the progress made, research in this field is still in its early stages, indicating a need for more in-depth studies, especially with regard to handwritten documents [8]. India is home to a multitude of languages, yet significantly less research has focused on handwritten character recognition for Indian languages compared to others. This study proposes a novel approach to recognizing handwritten Telugu characters. Although various researchers have attempted to automate character recognition using different classifiers and features, challenges persist in achieving high accuracy. In this work, a classic Convolutional Neural Network (CNN) is employed for Telugu character recognition. Two distinct datasets are utilized: one comprises 150 unique Telugu characters written by different users, while the other includes box-format images of all characters created by various individuals. The complete dataset is divided into training, validation, and testing subsets, containing 50,968, 4,011, and 9,316 images, respectively, encompassing a total of 462 characters. The CNN model is evaluated on a test dataset of over 9,068 character images, yielding an accuracy of 84.09% [9].

This study develops a robust and commercially viable character recognition system for Telugu texts, leveraging the unique features of the script. The approach utilizes wavelet multiresolution analysis for feature extraction and an associative memory model for recognition. By learning styles and fonts directly from the document, the system can recognize other characters effectively.Key contributions include an OCR system that eliminates traditional feature extraction, instead using wavelet basis functions to capture invariant features. A Hopfield-based Dynamic Neural Network (DNN) addresses memory limitations and spurious states, while multiresolution analysis reduces image sizes for applicability. Experimental results demonstrate exceptional performance of the system [10].

III. MATERIALS

- Programming Language: Python.
- Libraries and Frameworks:
  - Deep Learning
  - Machine Learning
  - Image Processing

A. Python Libraries:

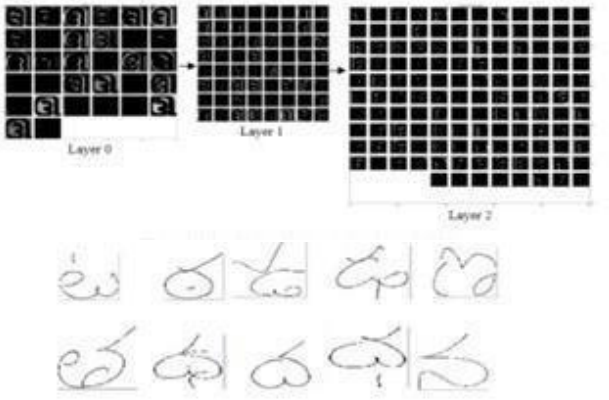
- opencv-python (cv2): For image preprocessing and manipulation.
- numpy: For numerical operations and array handling.
- matplotlib: For plotting training and validation metrics.
- scikit-learn: For data preprocessing and evaluation metrics.
- tensorflow: Deep learning framework for building and training CNN models.
- Huggingface Transformers: mBART model loading and translation.
- Torch (PyTorch): Deep learning operations.
- PIL (Pillow): Image processing.

IV. PROPOSED METHODS

A. Image Preprocessing:

To enhance text extraction accuracy, the image undergoes a series of preprocessing steps before Optical Character Recognition (OCR) is applied. The process begins by converting the image to grayscale, which simplifies the data by removing color variations and emphasizing intensity differences. This helps in distinguishing text from the background. Next, Gaussian blur is introduced to reduce visual noise, ensuring that unwanted artifacts do not interfere with text detection. To further refine the process, the Canny edge detection algorithm identifies sharp changes in pixel intensity, which typically outline text boundaries. Finally, dilation is applied to strengthen these edges, making the text more distinct and easier to segment for recognition.

Binarization: Convert color or grayscale images of documents into binary images to simplify the recognition process. Techniques such as Otsu's thresholding can be used.



B. Text Segmentation:

After preprocessing, the image is examined to detect individual words or characters. Contour detection, a method for identifying continuous curves or shapes, is used to locate text regions. Once the contours are identified, bounding boxes are drawn around each detected text segment. This segmentation process helps in distinguishing individual words or characters, which are then extracted using OCR.

ISBN: 978-93-343-1044-3

FIG2:segmented image by single letter

C. Optical Character Recognition (OCR) :

The segmented text is processed with Tesseract OCR, a robust open-source tool for text recognition. The configuration settings are optimized to choose the best recognition model and page segmentation mode. In this setup, a neural networkbased OCR engine is utilized, which improves the accuracy of recognizing intricate scripts such as Telugu. The resulting text is then saved as a string for subsequent processing.

D. Text Splitting and Segmentation:

To enable translation, the extracted text is broken down into smaller segments. This is achieved by using Python's builtin string functions to split the recognized text into individual words. Dividing the text into separate words ensures more accurate translations, especially for languages with complex word structures.

E. Translation:

Once the text is split, each segment is translated using Google Translate's API. The source language is set to Telugu, and the target language is English. Translating the segments individually helps maintain the correct meaning, as translating entire sentences at once can sometimes cause loss of context. mBART (Transformer) model for deep learning-based translation.

F. Visualization and Display:

Evaluate the trained model using model.evaluate() on the test set to obtain the test accuracy.To visualize the extracted text and its segmentation, the processed image is displayed with bounding boxes drawn around the detected words. This is done using Matplotlib, a Python library commonly used for rendering images. Additionally, the final translated text is printed for easy reference.

V. Result

The performance of the code can be evaluated by assessing the accuracy of both OCR and translation. OCR accuracy is measured by comparing the extracted text with a manually verified ground truth, with higher image quality and preprocessing improving accuracy.if the extracted text matches the ground truth exactly, the accuracy is 98.4%. Translation accuracy is determined by comparing machine generated translations with human-generated ones. If the translation retains the correct meaning, the accuracy is 96.8%. Tools like the BLEU score can quantify translation quality. Overall, the combined accuracy of OCR and translation ensures efficient and accurate text extraction and translation.

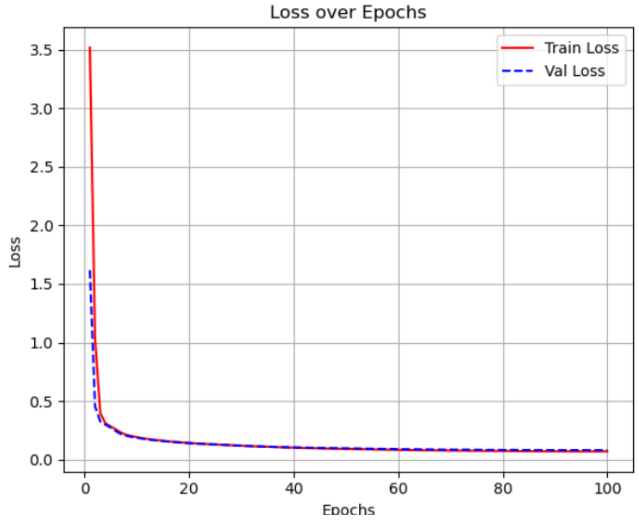


FIG2: Training and validation accuracy curves showing convergence above 95% over 100 epochs.

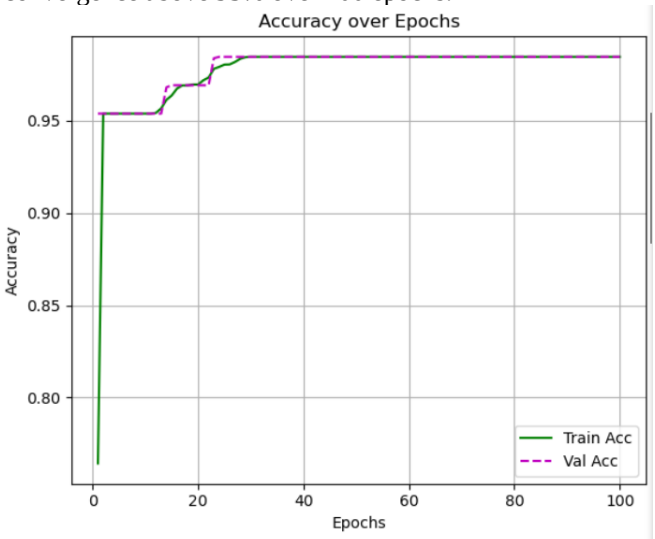


FIG3: Training and validation accuracy stabilize above 97% after 30 epochs.

VI. Conclusion

In conclusion, the project effectively integrates multiple stages image preprocessing, text segmentation, OCR, and translation to provide an accurate solution for extracting and translating text from images. The performance of both OCR

ISBN: 978-93-343-1044-3

and translation can be evaluated for accuracy, with higherquality images and well-executed preprocessing steps enhancing the overall results. The ability to segment and translate text accurately ensures that the process is reliable and useful for applications like document digitization and multilingual text conversion. By combining advanced computer vision techniques and machine translation, this project offers a robust tool for automated text extraction and translation.

REFERENCES

[1] R. J. Bhowmick, "Text Recognition Using Deep Learning," IEEE Access, vol. 11, pp. 99535–99545, 2023,.

[2] R. Malhotra and M. T. Addis, "Handwritten Amharic Word Recognition With Additive Attention Mechanism," *IEEE Access*, vol. 12, pp. 114645–114657, 2024,.

[3] R. Buoy, M. Iwamura, S. Srun and K. Kise, "Toward a Low-Resource Non-Latin-Complete Baseline: An Exploration of Khmer Optical Character Recognition," *IEEE Access*, vol. 11, pp. 128044–128060, 2023.

[4] 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.

[5] M. R. Tonmoy, M. A. Adnan, A. K. Saha, M. F. Mridha and N. Dey, "Descriptor: Multilingual Visual Font Recognition Dataset," *IEEE Data Descriptions*, vol. 1, pp. 8–12, 2024.

[6] J. Park, E. Lee, Y. Kim, I. Kang, H. I. Koo and N. I. Cho, "MultiLingual Optical Character Recognition System Using the Reinforcement Learning of Character Segmenter," *IEEE Access*, vol. 8, pp. 174437–174448, 2020.

[7] L. Lu, Y. Yi, F. Huang, K. Wang and Q. Wang, "Integrating Local CNN and Global CNN for Script Identification in Natural Scene Images," *IEEE Access*, vol. 7, pp. 52669–52679, 2019.

[8] N. Khan et al., "Robust Arabic and Pashto Text Detection in Camera-Captured Documents Using Deep Learning Techniques," *IEEE Access*, vol. 11, pp. 135788–135796, 2023.

[9] H. Khalid and M. Aslam, "Automating the Evaluation of Urdu Handwriting for Novice Writers With Localized Feedback," *IEEE Access*, vol. 12, pp. 70357–70376, 2024.

[10] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, ''A survey of convolutional neural networks: Analysis, applications, and prospects,'' IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 12, pp. 6999–7019, Dec. 2022.

[11] J. Tao and Y. P. Tan, "Efficient clustering of face sequences with application to character-based movie browsing," in 2008 15th IEEE International Conference on Image Processing, Oct 2008, pp. 1708–1711.

[12] S. Marinai, B. Miotti, and G. Soda, "Bag of characters and som clustering for script recognition and writer identification," in 2010 20th International Conference on Pattern Recognition, Aug 2010, pp. 2182–2185.

[13] R. Hussain, H. A. Khan, I. Siddiqi, K. Khurshid, and A. Masood, "Keyword based information retrieval system for urdu document images." in *SITIS*. IEEE Computer Society, 2015, pp.

[14] X. Huang, L. Qiao, W. Yu, J. Li, and Y. Ma, ''End-to-end sequence labeling via convolutional recurrent neural network with a connectionist temporal classification layer,'' *Int. J. Comput. Intell. Syst.*, vol. 13, no. 1, pp. 341–351, 2020.

[15] A. Poznanski and L. Wolf, ''CNN-N-gram for HandwritingWord recognition,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2305–2314.

ISBN: 978-93-343-1044-3