International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

A COMPLETE OCR MODEL WITH CUSTOMISED CNN ARCHITECTURE

SRINIVASA RAO DHANIKONDA[1], N SUBHASH CHANDRA[2]
[1]RESEARCH SCHOLAR, CSE, JNTUH, HYDERABAD, INDIA
[2]PROFESSOR, CSE, CVRCOE, JNTUH, HYDERABAD, INDIA

Abstract: Optical Character Recognition is one of the significant challenges for regional languages like Telugu, Tamil, Kannada, and Malayalam. OCR takes input as an image and tries to transfer that into an editable text by using different steps internally like segmentation, classification, and recognition. In this digital era, all primarily communicate using media, and 3.3 Million images are used per minute on average. Understanding these images is necessary, and sometimes it needs processing. So many models have been defined for the English language and achieved almost 100% accuracy. Compared to the English language, constructing the OCR model for the Telugu language is very difficult because of its structure. The Telugu language consists of vowels(V), consonants (C), and combinations of both C and V. The major challenge in the Telugu OCR is the segmentation of text regions. A dynamic segmentation technique used to solve this issue. To recognize the characters from the segmentation regions, a customized Telugu CNN model was introduced. This paper proposed a customized Telugu CNN model and compared the model with different architectures involved in the CNN architecture family.

Keywords: OCR (Optical Character Recognition), convolution neural networks (CNN), scripts, segmentation, classification.

Introduction

There was limited research in the maturation of a complete OCR program for Telugu script. While the access to a massive internet corpus of scanned files warrants the requirement to get the OCR system, the more complex script and agglutinative grammar create the issue hard. Constructing a system that works nicely on real-world files comprising sound and erasure is more complicated. The endeavor of OCR is principally divided into segmentation and recognition. That of another directs the plan of each. The stronger (to sound, erasure, skew, etc.) that the segmentation will be, the simpler the job of this recognizer becomes and vice-versa. The techniques utilized in segmentation are similar through areas. That is because, generally, one connected component (a neighboring area of ink) could be expressed as one unit of text or character. Although this principle applies to the Roman broadcasts with few exceptions, it doesn't hold complicated scripts such as Devanagari and Arabic. Phrases aren't letters; they have been written in a single contiguous slice of ink. The Telugu script consists of intermediate complexity, in which consonant-vowel pairs have been composed as a single unit. The recognition task is split to feature extraction and classification. The former was hand-engineered for a lengthy moment. They train multiple neural networks, and pre-classify an input image based on its aspect ratio and feed it to the corresponding network. It reduces the number of classes that each sub-network needs to learn. But this is likely to increase the error rate, as a failure in preclassification is not recoverable. The neural network employed is a Hopfield net on a down-sampled vectorized image.

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
IJARST
ISSN: 2457-0362

Later work on Telugu OCR primarily followed the featurization-classification paradigm. Combinations like ink-based features with the nearest class centroid (Negi, Bhagvati and Krishna, 2001); ink-gradients with nearest neighbours (Lakshmi and Patvardhan, 2002); principal components with support vector machines (Jawahar, Kumar and Kiran, 2003); wavelet features with Hopfield nets (Pujari et al., 2004) were used. More recent work in this field (Kumar et al., 2011) focuses on improving the supporting modules like segmentation, skew-correction and language modeling. While our work was under review, Google Drive added an OCR functionality that works for Telugu and many other world languages. Although its details are not public, it seems to be based on their Tesseract multilingual OCR system (Smith, 2007) augmented with neural networks.

Telugu is the official language used for communication in Andhra Pradesh and Telangana states and is one of the South Indian languages. More than 80 million people are using the Telugu language regularly for communication.

Todo data entry into systems the keyboard or the mouse used in the olden days. So many alternatives are available in this digital era like speech, bar code, QR code, etc. All these methods work with a common target of automatic identification.

Optical Character Recognition processes the document images and produces an editable format for further usage.

Gustav Tauschek received the first patent for the OCR in 1929. In 1950 the Department of Defense scientist David Shepard built the first OCR machine. The name of the machine was “Gismo” The main motto behind the invention of the OCR machine was to reduce the labor of retyping the documents. It saves a lot of time and effort to convert hard copies of document images into softcopy.

Initial stages attempts at OCR development have been successful in automatic reading and data entry of various Roman and Latin scripts. For the English language, the accuracy achieved almost 100%.

Compared to the English language, the Telugu language contains so many distinct characters like consonants (C), vowels(V), and compound characters of these two(CV) combinations.

Phases of OCR

OCR takes input as a document image and passes this input to several stages like preprocessing (binarization, skew correction), line, word, and character segmentation, feature extraction, recognition, and mapping of the respective text as part of the output. Every stage has its strength to produce an error-free output.

Binarization

Understanding the input is very important to get a proper solution. Separating unnecessary points from the input gives fast and perfect solutions. Preprocessing stage is the first step in the OCR engine. This stage handles the noisy data. After removing the noise, binarization takes the noise-free image and converts the grayscale image into a binary image.

The binary image consists of only two colors, either white or black. The image background is white, and the foreground is black and vice versa. The existing, compared to grayscale image operations, the binary image gives good results compared to grayscale images.

Binarization helps to identify the content of the image from its background. Image binarization uses Otsu’s threshold to convert grayscale images into binary images.

Noise Removal

Noise can often be found in scanned documents due to the printer, scanner

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

quality, age, and print quality. It is important to remove this noise from the image before processing it. Low-pass filtering the image is a common approach that can be used for processing later. A filter is required to remove as much noise from the signal as possible in order to reduce it.

Thinning

A process called skeletonization or thinning is where an object's representation is created that is one pixel wide. It preserves an object's connections and ends points (Gonzalez & Woods, 2002). To make an image easier to recognize and analyze, it reduces its information. This makes it easier to identify relevant features. Figure 3 shows an example image of a photo before and after thinning. There are many different existing thinning methods that have been created Hilditch is the most often used algorithm. There are many variations.

Skew Detection and Correction

A few degrees of tilt (skew) can be expected when a human or machine operator is feeding a document into the scanner. Text lines in a digital picture connect with horizontal directions at an angle known as the skew angle. A variety of skew estimating techniques exist. There are two types of skew estimation methods. The projection profile of a document is the first. Another one is based on the clustering of related neighbor components. Skew estimation can also be done using techniques based on the Fourier transform and Hough transform. Chaudhuri & Pal (1997) provide a comprehensive overview of the various skew correction methods. The projection profile is a popular method of skew detection. Horizontal projection profiles are A one-dimensional array in which each element represents the number black pixels per row. The horizontal projection profile is useful for documents that have horizontal text lines. It features a peaked that is equal in width to the character height, and valleys that are equal in width to the spacing between lines. Since scan lines align with text lines, the projection profile is at the correct skew angle. It has a maximum height peak for line spacing valleys and text.

Segmentation

The next basic building block of the OCR engine is Segmentation. In this stage, individual character regions are isolated from the binary image. Much research work has been done so far some of the best outcomes are discussed in the literature study. Line, word, and character segmentations are segmented sequentially in the proposed work. In this stage, the number of lines, words and characters were computed.

Word region segmentation

English has a well-developed segmentation system for documents. Maximally stable extremal regions are used to separate characters in English in many papers. MSER cannot directly be applied to Telugu because most of the dheergams are separate. Minor changes were made to MSER in order to account for vatus and dheergas.

Character level segmentation

The Connected Components algorithm in Image Processing allows us to segment every character of the word. After the image has been binarized, the algorithm is used to separate the letters and vattus. The components are groups of binary pixels that contain the letters and vattus. The components are also stripped of little blobs. Some vattus in Telugu do not connect to the base letter. We measured the overlap distance in horizontal and vertical directions to connect the base letter and its vattu and then grouped them.

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

Literature survey

Shobha Rani N. et al. (2015) the proposed algorithm for text line segmentation of Telugu document images consists of three significant steps. The first step generates a fringe map. In the second step, Peak fringe numbers (PFNs) are located in the fringe map. The PFNs between text lines are determined by performing a filtering operation. Identifying PFNs that belong to an adjacent line and generating a segmenting path is not easy because the filtering operation leaves gaps. Hence, a broad region is constructed to cover the consonant moodier of a line and vowel modifiers of the following line (the overlapping and touching components of adjacent text lines). These regions cluster the PFNs between adjacent lines. In the last step, a segmenting path between lines is generated by joining the region's PFNs. Raashid MALIK et al. (2007) our initial objective is, therefore, to find and isolate the text in a scene. From a practical perspective, an extension of this work can lead to machine reading of highway signs such as exits, speed limits or cautions. It may also make barcoding superfluous since machines would able to read labels on merchandise directly. As graphical, textual information has been increasing; text extraction is a necessary procedure for recognition steps. Even though many approaches have been addressed now, the majority of them cannot solve text extraction problems. What surroundings make it complicated? Variants of font, style, size, special symbol, multilingual environment and performing on the binary images always hinder us from exploring the final gist. Thuswe propose the scheme to root out the above restrictions using an image, based on edge detection, histogram and width to height ratios, of which input is grey images, not binary. We have shown expected results by experiments, font style,size language independently, and text embedded into the background image.

Vasudev T. et al. (2016) proposed a technique for feature extraction and classification of Telugu handwritten script based on customized template matching approach to support caching technique for better performance. The caching technique is implemented using the central database with a cache database, maintaining the frequently used character templates for a set of all character templates. The XML database is used for defining the classes for various character templates, and the class representations are provided using a novel class structure designed based on XML tags. The proposed system exhibits the recognition efficiency o Otsu's n our test dataset with an overall accuracy of 83.55% for handwritten characters. The feature extraction by shape matching in conjunction with correlation-based classification has provided satisfactory results. The inclusion of REQUIRED_SIMILARITY_MEASURE to find a suitable match between the test template and professional template significantly reduces the conventional template matching technique's worst-case time complexity. However, there are few cases of misrecognition, especially for some of the confusing character pairs [ఎ , ఏ] [ఐ , ఒ] [ఒ , ఓ] etc. The confusing characters possess minor differences in their structural orientation, but the use of REQUIRED_SIMILARITY_MEASURE improves the template matching algorithm's performance. However, the design of an efficient post-processing methodology can correct the recognition errors with prime regard to confusing character pairs. This suffices the reliability of the system, which is currently under investigation. In addition to this, the technique of caching improves the overall performance of the classification approach. The proposed system's experimental results still improved by replacing template matching with an efficient feature

extraction technique to reach high recognition accuracy.

Optical character recognition (OCR) has been among the most studied issues in pattern recognition. However, the achievement of CNN's motivated us to utilize them for Telugu character recognition. The first recorded work on OCR to get Telugu could be dated back as early as 1977 from Rajasekharan; also, Deekshatulu utilized features that synthesize the curves that follow a letter also compare that this encoding using a group of predefined templates [12]. It managed to spot 50 primitive features also suggests that a two-stage syntax-aided character recognition program. The first effort to use neural networks first created with M.B. Sukhaswami et al., which compels several neural systems and pre-classifies a picture based on its characteristic ratio. It then feeds it into the corresponding system [17]. It revealed that the robustness of a Hopfield system to understand noisy Telugu characters. Afterwards, work on Telugu OCR mostly adopted closely by the feature classification paradigm. Jawahar et al. [14] describe a bilingual Hindi-Telugu OCR for documents containing Hindi and Telugu text. It is based on Principal Component analysis followed by support vector regression. They report an overall accuracy of 96.7% over an independent test set. They perform character level segmentation offline by their data collecting tools. However, they have only considered 330 distinct classes.

The work by Rakesh and Trevor [5] on Telugu OCR using convolutional neural networks is also fascinating. They used 50 fonts in four styles for training data, each image of size 48x48. However, they not consider all possible outputs (only 457 classes) of CNN. Kunte and Samuel work on Kannada OCR employs a two stage classification system similar to our approach. They have first used wavelets for feature extraction and then two-stage multi-layer perceptrons for the task of classification. They have divided the characters into separate subclasses but have not considered all possible combinations. For Telugu text in printed form, Arun K Pujari et al. [15] in 2002 proposed an OCR system. Text is scanned in the form of a grayscale image. Horizontal and vertical projection techniques are used for line and word segmentation. The zero-padding technique is used to convert characters into a fixed size. Wavelet analysis is used for obtaining information of images at different scales like 32x32. Performed 2-dimensional filtering so that 32x32 image is converted into 4, 8x8 images, which gives the average image. Then by using thresholding, convert images to binary which gives 64 bits, and these are referred to as signature of the input symbol.
For recognizing symbols, Dynamic Neural Network is used in which every node in the network is the Hopfield network. This method does not depend on font and shape. Some symbols dha, dhaa, na, and this technique does not correctly recognize sa. C. Vasantha Lakshmi et al. [16] proposed an OCR system in 2003 for printed text in Telugu. The scanned image is converted to a binary scale, and noise is removed through rectification. Skew is corrected, and then lines, words and symbols are extracted from text segmentation. Pre-Classification of each symbol by size property to compute real-valued direction features. Neural recognizers are used for classification, and finally, information associations of basic symbols for a word are outputted. Testing is performed on one lakh symbols, which resulted in 99% accuracy for DeskJet prints and laser prints using additional logic. OCR system for printed characters in Telugu was proposed by Negi et al. [13]

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
IJARST
ISSN: 2457-0362

in 2003. Nonlinear normalization is performed using a modified crossing count, which enhances the features of the input image. In different zones, pixel densities are used for searching the initial candidate of input glyph. If the candidates are found in-conclusive, they are passed through another stage where input image cavities are analyzed. Template matching is done based on Euclidean distance on normalized characters for nonlinear shapes which are controllable. This technique obtained correct results for 1463 glyphs out of 1500 glyphs which are collected from the magazine.

Many algorithms are available for spell-checking systems—the latest techniques developed for Indic scripts discussed in this section. Dictionary lookups and Statistical methods used for spell checking and correction for the Punjabi language discussed by Baljeet Kaur [1]. J. Bharathi, P. Chandrasekar Reddy [2] proposed combining script-level properties and structural properties to identify partial touching characters. Unicode approximation Model (UAM) introduced by N Shobha Rani [3]; using UAM, segmentation and preprocessing errors were solved and achieved 96% accuracy. Grigori Sidorov [4] used Tree Edit Distance (TED) for computing text similarity and, by using edit mapping, swapped misspelled words with suitable words. To extract noisy Telugu script images, K Mohana Lakshmi [5] proposed a SURF descriptor method. Using the word spotting technique Nagasudha D [6] described a keyword substitute method for framing words in Telugu document images. The detection and correction of errors in OCR were discussed by V S Vinitha [7] by using statistical language model (SLM) and dictionary-based methods. Instead of the Unicode form, the Akshara method produces good results. M Priya [8] proposed a "Hybrid optimization algorithm using N-gram based edit distance, to handle rule generation. This hybrid algorithm produced good results compared to the N-gram model and Edit distance model. By using "Segmentation Edit Distance (SED)" Daniel Pucher [9], measured the distance between two words.

Customized Telugu CNN model to extract the Telugu text from image document

Many algorithms are defined to extract Telugu text from text images. MSER (Maximally Stable External Regions) used for OCR English. By using the MSER algorithm, English characters are extracted with good accuracy. However, the same procedure is not suitable to extract Telugu characters because of dheeragam and vattus. To solve this issue, proposed a Line and Character Segmentation (LCS) algorithm.

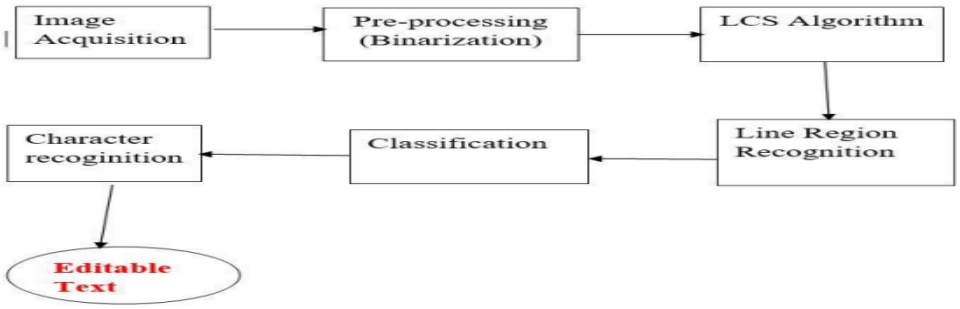

Figure 1: Basic OCR model with LCS

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

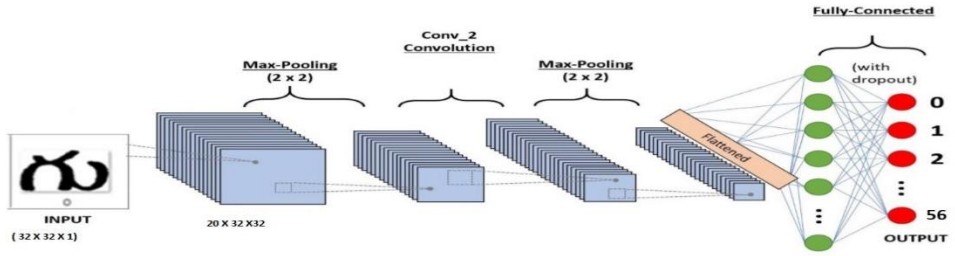

Figure 2: Basic CNN Architecture

The above figure illustrates the LCS Algorithm's working flow, as an initial step LCS algorithm takes the given image as input in the first phase. In this phase, the image's noise will be removed and convert the noise-free image into image binarization; here, the actual image will be converted to black and white pixels only. This will act as input to the LCS (Line and Character Segmentation). After applying the LCS algorithm, the number of lines and line region segmentations will be calculated. Character regions will be finalized from line region segmentations. Finally, editable Telugu characters will be produced by using classification algorithms.

To classify the characters, it takes input as character regions; based on the region's content by using classification techniques, the respective Telugu characters will be converted into editable text. The significant steps to identify character regions are listed below:

(i) Image acquisition
(ii) Word region segmentation
(iii)Character region segmentation
(iv)Classification

Line region segmentation is introduced instead of word region segmentation to extract Telugu characters more precisely in the proposed technique. In this proposed technique, the Telugu text image will act as input to the algorithm. After taking information, convert the image into binarization to avoid the noise; in binarization, the color image converted to binary form, i.e., either black or white pixels only. After the binarization step, convert the binary image into histograms. Based on the pixel levels, calculate the Hmax and Hmin positions. Now construct the top and bottom lines of the text image. Scan from the Hmax to Hmin, identify the gaps among vertical level histograms, based on the gaps, draw a horizontal line that divides one row to another row. The number of lines will be generated after reaching Hmin position. Calculate the distance between the adjacent bars. Find the mean of line distances. Identify very nearest line positions. Based on that, combine the nearest line positions. After connecting the most relative lines, the line positions will change dynamically so that dheergas and vattus regions are covered in every line. With this step, Line region segmentation covers the content of that respective line without any ambiguity. Repeat the same procedure horizontally to recognize the character regions. These character regions are supplied to the

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

CNN classifier to extract editable Telugu characters. To improve the accuracy, we built a new customized Telugu CNN architecture and achieved an accuracy of 98.9%.

Character Region Algorithm

1) Binarize the image (I)

$I_{binary} \leftarrow I$

2) Construct Horizontal Pixel Projection and Segment lines

a) Calculate the sum of values of each row

$$HPP(i) = \sum_{j=1}^{N} I(i,j)$$

b) Compute number of line regions

$$th\_lines = \left(\sum_{j=1}^{N} I(i,j)\right)/N \quad // \; N - number\ of\ rows$$

$$line\_counter\ (lc) = 0$$

$$lc = \begin{cases} lc+1 & HPP[i] = 0 \ and \ HPP[i+1] > 1 \ or \ HPP[i] > 1 \ and \ HPP[i] = 0 \\ lc & otherwise \end{cases}$$

c) return lc

3) Construct vertical pixel projection (VPP)

$$VPP(i) = \sum_{i=1}^{M} I(i,j) \quad // \; M\text{-}number\ of\ columns$$

4) Compute number of character regions

$$th\_char = \left(\sum_{i=1}^{M} I(i,j)\right)/M$$

$$char\_counter\ (cc) = 0$$

$$cc = \begin{cases} cc+1 & VPP[i] = 0 \ and \ VPP[i+1] > 1 \ or \ VPP[i] > 1 \ and \ VPP[i] = 0 \\ cc & otherwise \end{cases}$$

$$CR_k = \begin{cases} start & VPP[i] = 0 \ and \ VPP[i+1] != 0 \\ end & VPP[i] != 0 \ and \ VPP[i] = 0 \end{cases}$$

where CR is character region and k is 1,2,3 ...

5) return CR

6) Recognition of Characters
- Send each CR to the CNN and return respective class.

Figure 3: LCS Algorithm

CNN Architecture	Accuracy
MC Cifar	**98.60**
MC Lenet	**98.62**
TCCNN-S	**97.95**
Proposed	**98.9**

Table 1: Comparison of different models with the proposed model

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

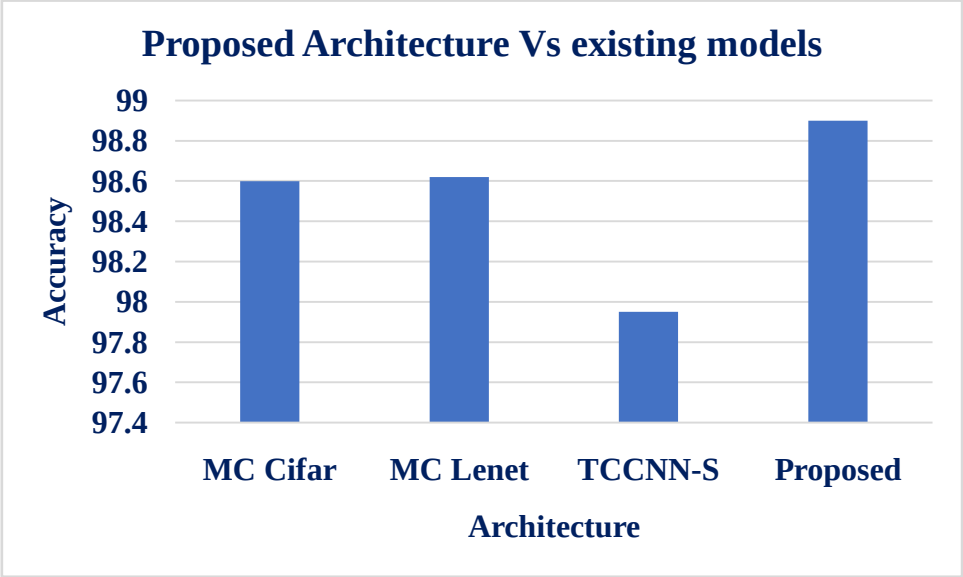

Figure 4: Comparison of Existiling models with proposed architecture

Conclusion and Future Direction

We have presented a procedure for extracting the Telugu characters from text images, which is challenging compared to other languages. We proposed a framework to solve the major issues in existing algorithms. LCS algorithm produces the dynamic line region segmentation, which leads to extract Telugu characters efficiently. Segmentation and classification are significant challenges in recognition of Telugu characters. We studied different frameworks for segmentation and classifications. Still, there is a need to improve the segmentation algorithms to get more accuracy.

References

[1] Babu, Arja Rajesh. "OCR for Printed Telugu Documents." Diss. Indian Institute of Technology Bombay Mumbai, 2014.

[2] Hu, Peifeng, et al. "Recognition of gray character using gabor filters." Information Fusion, 2002. Proceedings of the Fifth International Conference on. Vol. 1. IEEE, 2002.

[3] Ramanathan, R., et al. "Robust feature extraction technique for optical character recognition." Advances in Computing, Control, & Telecommunication Technologies, 2009. ACT'09. International Conference on. IEEE, 2009.

[4] Rao, P. V. S., and T. M. Ajitha. "Telugu script recognition-a feature based approach." Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on. Vol. 1. IEEE, 1995.

[5] Achanta, Rakesh, and Trevor Hastie. "Telugu OCR Framework using Deep Learning." arXiv preprint arXiv:1509.05962 (2015).

[6] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.

[7] Bastien, Frédéric, et al. "Theano: new features and speed improvements." arXiv preprint arXiv:1211.5590 (2012).

[8] Cire¸san, Dan, and Ueli Meier. "Multi-column deep neural networks for offline handwritten Chinese character classification." Neural Networks (IJCNN), 2015 International Joint Conference on. IEEE, 2015.

[9] Otsu, Nobuyuki. "A threshold selection method from gray-level histograms." IEEE

transactions on systems, man, and cybernetics 9.1 (1979): 62-66.
[10] Wolf, Christian, J-M. Jolion, and Francoise Chassaing. "Text localization, enhancement and binarization in multimedia documents." Pattern Recognition, 2002. Proceedings. 16th International Conference on. Vol. 2. IEEE, 2002.
[11] Karatzas, Dimosthenis, et al. "ICDAR 2015 competition on robust reading." Document Analysis and Recognition (ICDAR), 2015 13th International Conference on. IEEE, 2015.
[12] Rajasekaran, S. N. S., and B. L. Deekshatulu. "Recognition of printed Telugu characters." Computer graphics and image processing 6.4 (1977): 335-360.
[13] Negi, Atul, Chakravarthy Bhagvati, and B. Krishna. "An OCR system for Telugu." Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on. IEEE, 2001.
[14] Jawahar, C. V., MNSSK Pavan Kumar, and SS Ravi Kiran. "A bilingual OCR for Hindi-Telugu documents and its applications." Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on. IEEE, 2003.
[15] Arun K Pujari, C Dhanunjaya Naidu, BC Jinaga, "An Adaptive Character Recognizer for Telugu Scripts using Multiresolution Analysis and Associative Memory ", ICVGIP, Ahmedabad,2002.
[16] C. Vasantha Lakshmi, C. Patwardhan, "High accuracy OCR system for printed Telugu text", TENCON 2003, Conference on Convergent Technologies for Asia-Pacific region, Volume 4,15-17 October.
[17] M Swamy Das and Ram Mohan Rao, "Evaluation of Neural Based Feature Extraction Methods for Printed Telugu OCR System ", Advances in Computer Science and Information Technology, Volume 2, 11, 2015.
[18] N. Shobha Rani, Vasudev T, "A performance efficient technique for recognition of Telugu script using template matching", I. J. Image, Graphics and Signal Processing, Volume 8, 2016.
[19] C. Vasantha Lakshmi, Ritu Jain, and C. Patvardhan, 2006. OCR of Printed Telugu Text with High Recognition Accuracies. ICVGIP, springer, 786- 795.
[20] Chandra Prakash, Konkimalla & Srikar, Y. & Trishal, Gayam & Mandal, Souraj & Channappayya, Sumohana. (2017). Optical Character Recognition (OCR) for Telugu: Database, Algorithm and Application.
[21] Babu, Arja Rajesh. OCR for Printed Telugu Documents. Diss. Indian Institute of Technology Bombay Mumbai, 2014.
[22] Achanta, Rakesh, and Trevor Hastie. "Telugu OCR Framework using Deep Learning." arXiv preprint arXiv:1509.05962 (2015).
[23] Rajasekaran, S. N. S., and B. L. Deekshatulu. "Recognition of printed Telugu characters." Computer graphics and image processing 6.4 (1977): 335-360.
[24] P. Vithlani and C. Kumbharana, "A study of optical character patterns identified by the different ocr algorithms," International Journal of Scientific and Research Publications, vol. 5, no. 3, pp. 2250–3153, 2015.
[25] N. A. Jebril, H. R. Al-Zoubi, and Q. A. Al-Haija, "Recognition of handwritten arabic characters using histograms of oriented gradient (hog)," Pattern Recognition and Image Analysis, vol. 28, no. 2, pp. 321–345, 2018.
[26] D. Lin, F. Lin, Y. Lv, F. Cai, and D. Cao, "Chinese character captcha recognition and performance estimation via deep neural network," Neurocomputing, vol. 288, pp. 11– 19, 2018.
[27] R. Verma and J. Ali, "A-survey of feature extraction and classification techniques in ocr systems," International Journal of Computer Applications & Information Technology, vol. 1, no. 3, pp. 1–3, 2012.

[28] Kaur, Baljeet. "Review on error detection and error correction techniques in NLP." Int. J. Adv. Res. Comput. Sci. Software Eng 4 (2014): 851-853.
[29] Bharathi, J., and P. Chandrasekar Reddy. "Segmentation of Touching Conjunct Consonants in Telugu using Minimum Area Bounding Boxes." Int. J. Soft Comput. Eng 3.3 (2013): 260-264.
[30] Rani, N. Shobha, and T. Vasudev. "Post-processing methodology for word level Telugu character recognition systems using Unicode Approximation Models." 2015 International Conference on Trends in Automation, Communications and Computing Technology (I-TACT-15). IEEE, 2015.
[31] Sidorov, Grigori, et al. "Computing text similarity using tree edit distance." 2015 Annual Conference of the North American Fuzzy Information Processing Society (NAFIPS) held jointly with 2015 5th World Conference on Soft Computing (WConSC). IEEE, 2015.
[32] Lakshmi, K. Mohana, and T. Ranga Babu. "Searching for Telugu script in noisy images using SURF descriptors." 2016 IEEE 6th International Conference on Advanced Computing (IACC). IEEE, 2016.
[33] Lakshmi, K. Mohana, and T. Ranga Babu. "A new hybrid algorithm for Telugu word retrieval and recognition." nternational Journal of Intelligent Engineering and Systems 11.4 (2018).
[34] Vinitha, V. S. Error detection and correction in Indic OCRs. Diss. International Institute of Information Technology Hyderabad, 2017.
[35] Priya, M., R. Kalpana, and T. Srisupriya. "Hybrid optimization algorithm using N-gram based edit distance." 2017 International Conference on Communication and Signal Processing (ICCSP). IEEE, 2017.
[36] Pucher, Daniel, and Walter G. Kropatsch. "Segmentation edit distance." 2018 24th International Conference on Pattern Recognition (ICPR). IEEE, 2018.
[37] Sindhu, D. V., and B. M. Sagar. "Dictionary based machine translation from Kannada to Telugu." IOP conference series: materials science and engineering. Vol. 225. No. 1. IOP Publishing, 2017.
[38] Chakravarthi, Bharathi Raja, et al. "Bilingual lexicon induction across orthographically-distinct underresourced Dravidian languages." (2020).
[39] Soujanya, B., and T. Sitamahalakshmi. "Optimization with ADAM and RMSprop in Convolution neural Network (CNN): A Case study for Telugu Handwritten Characters." International Journal 8.9 (2020).
[40] Singh, Shashank, and Shailendra Singh. "HINDIA: a deep-learning-based model for spell-checking of Hindi language." Neural Computing and Applications 33.8 (2021): 3825-3840.
[41] Prasad, Palanati Durga, K. V. N. Sunitha, and B. Padmaja Rani. "Word N-gram based approach for word sense disambiguation in Telugu natural language processing." Int. J. Rec.