

Research Article

An Efficient Deep Learning Model with Interrelated Tagging Prototype with Segmentation for Telugu Optical Character Recognition

Srinivasa Rao Dhanikonda,¹ Ponnuru Sowjanya,¹ M. Laxmidevi Ramanaiah,² Rahul Joshi ,³ B. H. Krishna Mohan,⁴ Dharmesh Dhabliya,⁵ and N. Kannaiya Raja ⁶

¹Department of Computer Science Engineering, GITAM Deemed-to-be-University, Hyderabad, India

²Department of Electrical and Electronics Engineering, Institute of Aeronautical Engineering, Hyderabad, India

³Department of Computer Science Engineering, Symbiosis Institute of Technology, Pune, Symbiosis International (Deemed University), Pune, India

⁴Department of Information Technology, RVR & JC College of Engineering, Guntur, India

⁵Department of Computer Engineering, Vishwakarma Institute of Information Technology, Pune, Maharashtra, India

⁶Department of Computer Science, IOT-HH Campus Ambo University, Ethiopia

Correspondence should be addressed to N. Kannaiya Raja; kannaiya.raja@ambou.edu.et

Received 14 June 2022; Revised 24 July 2022; Accepted 5 August 2022; Published 29 August 2022

Academic Editor: Punit Gupta

Copyright © 2022 Srinivasa Rao Dhanikonda et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

More than 66 million people in India speak Telugu, a language that dates back thousands of years and is widely spoken in South India. There has not been much progress reported on the advancement of Telugu text Optical Character Recognition (OCR) systems. Telugu characters can be composed of many symbols joined together. OCR is the process of turning a document image into a text-editable one that may be used in other applications. It saves a great deal of time and effort by not having to start from scratch each time. There are hundreds of thousands of different combinations of modifiers and consonants when writing compound letters. Symbols joined to one another form a compound character. Since there are so many output classes in Telugu, there's a lot of interclass variation. Additionally, there are not any Telugu OCR systems that take use of recent breakthroughs in deep learning, which prompted us to create our own. When used in conjunction with a word processor, an OCR system has a significant impact on real-world applications. In a Telugu OCR system, we offer two ways to improve symbol or glyph segmentation. When it comes to Telugu OCR, the ability to recognise that Telugu text is crucial. In a picture, connected components are collections of identical pixels that are connected to one another by either 4- or 8-pixel connectivity. These connected components are known as glyphs in Telugu. In the proposed research, an efficient deep learning model with Interrelated Tagging Prototype with Segmentation for Telugu Text Recognition (ITP-STTR) is introduced. The proposed model is compared with the existing model and the results exhibit that the proposed model's performance in text recognition is high.

1. Introduction

The increasing use of resources such as paper documents, pictures, smart phones, and ipads has made online Telugu text recognition one of the most active and demanding research fields in the realm of pattern recognition [1]. Southern India is home to the Telugu people, who speak Telugu as their native tongue. It has a vowel count of 16 and a

consonant count of 36 [2]. There are many Telugu characters that are used in Indian languages to denote the beginning or end of a single word [3]. Telugu characters can be one of two types: either a pure vowel or a combination of vowels and consonants. There are therefore 52 vowels and consonants in all in the English language [4]. Character identification is more difficult in some languages than others because of the varying shapes, strokes, and numbers of characters in each

language. The number of native users of Telugu places it third in India. Telugu has a large number of characters that have the same shape [5].

There is a growing need for software that can read handwritten characters when they are fed to a computer these days. These days, there is a huge need for archiving information found in books or other papers into a computer, where it can be found again via a search engine in the future. This can be accomplished by scanning the data that needed to be saved and stored. Problem is that we cannot search the image's information [6], therefore, it may be tough to read the text. This is a problem because handwritten font characters differ from computer system font characters. As a result, when the computer attempts to decipher the characters [7], it fails to do so. Document processing is the practise of storing paper documents' content in a computer storage location, then reading, searching, and sorting through it. It is necessary to process information relating to English and other languages from time to time in document processing [8]. It is necessary to use a character recognition software system to process the documents. The character gesture models of Telugu characters are shown in Figure 1.

Document Image Analysis (DIA) is another name for this procedure. This is why it is important to work on character recognition software that can automatically convert documents or images into an editable version using DIA. While there are numerous ways to go about this task, the Optical Character Recognition method is used. Visual patterns into alphanumeric or other characters are an aim of optical character recognition [9]. An OCR software engine reads the machine-printed text or handwritten script contained in the digital image and converts it into an editable computer digital text format that can be edited by the user. Due to the lack of data sets and training deep convolutional neural networks [10], it is unable to recognise Telugu handwritten letters even though it can easily get information from diverse images. Handwritten, typed, or reading the passages can be converted to machine-encoded [11] text using OCR. OCR technology attempts to solve the difficulty of identifying a wide range of characters in scanned documents or images [12]. To make handwritten or printed data machine-readable or digitally stored images, the characters must be recognised and transformed [13]. OCR allows us to digitise unique identifiers or codes made up of numbers and letters. The Telugu character representation is shown in Figure 2. The top stroke is the tallest glyph in the font's maximal height at a particular text size above the baseline. The bottom stroke is the lowest glyph in the font's lowest position for a specific text size that can be below the baseline maximally.

Convolutional Neural Networks (CNNs) are employed in a wide range of pattern recognition applications, including the analysis of medical images, paper documents, pictures, touch displays, and other devices [14]. Using CNNs for character recognition [15] both online and offline is a viable option. Earlier attempts at Telugu text handwritten recognition are one area to look into; offline recognition in Indian languages is another and research for greater accuracy, i.e., greater than 90%, is the final area of focus [16]. This

అ	ఆ	ఇ	ఈ	ఉ	ఊ	ఋ
a	ā	i	ī	u	ū	ṛ
[ʌ]	[a:]	[i]	[i:]	[u]	[u:]	[ri/ru]
ఋ	ఎ	ఏ	ఐ	ఒ	ఓ	ఔ
ṛ	e	ē	ai	o	ō	au
[ri/ru]	[e]	[e:]	[aj]	[o]	[o:]	[aw]

FIGURE 1: Telugu character gestures.

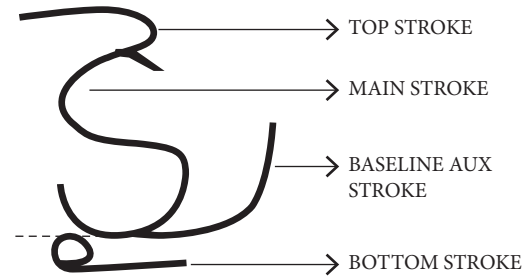


FIGURE 2: Telugu character representation.

work examines the recognition of Telugu character recognition algorithms with a high-recognition accuracy of over 90% and minimal training time. A CNN model was put forth for the segmentation and classification of grey scale images. The authors trained two CNN models for this purpose, which has the disadvantage of requiring more computing time. In South India, Andhra Pradesh, and Telangana, Telugu is the most commonly spoken Dravidian language. The data set includes Telugu characters, diacritics, and scripts. Vowel diacritics, independent vowels, consonants, and consonant modifiers make up the bulk of the data collection. Some consonant-vowel combinations are difficult to separate.

Using a computer to manipulate digital photographs for the purpose of analysing and retrieving data is referred to as the DIP process. There are many different ways to process digital photos because digital images come from a wide range of electromagnetic spectrum sources. Digital image processing methods can be divided into three levels: low, medium, and high. Low-level approaches cover a wide range of processes used to get images ready for further processing when needed. Feature analysis and presentation are performed on the preprocessed images [17] at the middle level of processing, which involves the extraction of essential image components. Further interpretation of recognised objects is dealt with at the higher level of processing.

Image processing has a wide range of applications since the need arises from a wide range of applications in several fields. Medical image processing, biometric image processing, satellite image processing, and document image processing are all examples of digital image processing applications. It is believed that computer vision is a continuation of deep learning and its methods are comparable to human vision tasks [18]. Manual techniques used to conduct

real-time activities requiring intelligence are usually achieved artificially with the aid of machinery. The goal of computer vision techniques is to automate human tasks such as reading, understanding, visualising, interpreting, and introspecting [19]. Humans do daily activities such as reading, learning, analysing, and interpreting by using a variety of examples. To achieve pattern recognition, these examples are fed to machines using a variety of machine learning methods [20]. Recognition of patterns in digital images may be done quickly and cheaply with pattern recognition software since it deals with detection, categorisation, and recognition of different patterns accessible in digital images [21]. All of these jobs, taken together, create an automated activity and are carried out using machine learning processes. Images are segmented according to their traits and qualities in the digital image processing branch known as image segmentation [22]. Image segmentation is the process of dividing a visual image into segments with comparable characteristics [23]. Image objects are the segments of the image into which images are divided [24]. When homogenous pixels are defined into spectrally comparable image segments, segmentation takes place [25]. This method aims to transform the image's properties into something more meaningful, which makes interpretation and classification easier and faster.

2. Literature Survey

Velpuru et al. [1] suggested an OCR system for Telugu text printed on paper. Grey scale images of text are generated during the scanning process. Line and word segmentation are accomplished by the use of horizontal and vertical projection techniques. In order to make characters a set size, the zero-padding approach is employed. In order to retrieve visual information at various sizes, such as 32×32 , wavelet analysis is utilised. Two-dimensional filtering was used to reduce a 32×32 image to four 8×8 images, resulting in an average image. Next, apply thresholding to images to make them binary, giving you 64 bits of information called the signature of the input symbol. Every node in the network is a Hopfield network in order to recognise symbols using Dynamic Neural Networks. Font and form are not an issue with this approach. There are a few symbols that this method fails to recognise correctly.

Bhagvati et al. [3] suggested a Telugu OCR system. To reduce noise, the scanned image is rendered in binary scale and then rectified. After the skew is fixed, the text segmentation can be used to extract individual lines, words, and symbols. Pre-classification of each symbol based on its size in order to derive real-valued direction characteristics. Classification is carried out using neural recognisers, and information on the basic symbol associations for a word is then output. DeskJet prints and laser prints with additional logic were tested and found to be 99 percent accurate. Prakash et al. [5] proposed an OCR system for Telugu printed characters. The improved crossing count method of nonlinear normalisation improves the features of the input image. The initial candidate of the input glyph is searched for using pixel densities in various zones. The second round of

analysis of the input image cavities is performed on candidates who are deemed inconclusive at the first. For non-linear shapes that can be controlled, template matching is done using Euclidean distance on normalised characters.

Nagarajan et al. [6] developed an OCR. Scanned documents are processed by applying filters, converting to binary, and using projection profiles to correct skew. Word and line segmentations are conducted on the text blocks that have been taken from the pages. For Hindi, it is necessary to remove Shirorekha, whereas for Telugu, it is necessary to separate the related components. To do feature extraction, all components have to be scaled to a set scale. The full image must be used as a feature vector to improve the results. To make the feature vector smaller and more manageable for diverse languages and handwritten manuscripts, principal components can be employed, as they are font-independent and universal.

Karthick et al. [12] created an OCR system for printed Telugu text. Instead of matching words with a dictionary, which adds to the computing cost, they employed the edge histogram and a confusion table to eliminate confusion between similar symbols. When converting a grey scale image to a binary image, thresholding is employed. Skew detection and skew correction are both accomplished with the help of modified Hough Transforms. Profiling was used to separate the text in the image into words and lines. It uses the Nearest Neighbor (NN) classifier system and a preliminary classification strategy to identify a symbol. Confusion tables are used to avoid misinterpretations due to scanning noise or paper imperfections, which can cause symbols that look identical to be mistakenly recognised. It recognises fonts of all sizes and has a 91.5 percent recognition rate for all typefaces. If there are any symbols in the set that are causing confusion, the logic to rectify it will only run when the recogniser determines that the symbols are present.

An OCR for printed Telugu characters was proposed by Benita Galaxy et al. [16]. There are three stages to the process. To begin, gather as much information as possible to train the algorithm. To differentiate characters from words, consider using the vertical gaps. Then, extract the feature vectors for each character. To fix the font size, Otsu's threshold approach is used to transform the text image to binary. To obtain an image matrix from border pixels, the Hilditch technique is employed. Supervised learning is used to train an artificial neural network. It was created to deal with large quantities of standardised documents. Darshan et al. [17] proposed an OCR method for printed Telugu text. To deal with the broken characters, the classifier makes use of feedback and character segmentation to increase the accuracy of the OCR system by using orthographic features in Telugu. Binary adaptation and skew detection are two of the tools employed in this process. Line segmentation, word segmentation, and character segmentation are all conducted based on the projection profiles.

An OCR system was proposed by Magesh [10]. There are not many datasets sets for south Indian languages, so they made their own by gathering documents from places like schools and law offices. Binarisation was accomplished by utilising *Otus*' approach. Noise is eliminated through the use

of the morphological opening. Image digits are manually segmented before being transformed to 32×32 pixels. To get the features, a normalised image divided into 8×8 squares is performed. Features are recognised for each zone by estimating the densities of pixels. SVM and KNN classifiers are used in the classification process. Bilingual samples have recognition accuracy of 92% and 94%, respectively. Classifier improvement is required to boost recognition rates even further.

Using an OCR system to identify the font size and size of printed Telugu text was proposed by Natrayan et al. [20]. To begin, the image is transformed from grey scale to binary and then cleared. To the top and bottom of the screen, add 20 rows of white pixels, and the left and right columns, 20 rows of white pixels. The horizontal profile of each row is calculated and used to segment the lines. Determine the horizontal profiles of the head, top, bottom, and baselines. Components that are linked are decoupled. Using Zonal information, each Component is divided into two categories: core and noncore. The noncore components have a tick mark next to them. Compare the two ratios above to determine the pixel ratio, aspect ratio, and font size.

Computer Graphics Image Processing has been presented by Rajnoha et al. [21] for the recognition of written Telugu characters. To the best of our knowledge, this was the first study to look at Telugu character OCR. A two-stage preprocessor character recognition method is proposed, which finds 50 primitive properties. The primitive shapes are detected and removed using a knowledge-based search in the first stage. The pattern is coded by tracing along points on it in the second stage after primitives have been removed. A decision tree is used to classify data. Individual characters are defined by the right combination and overlay of primitives.

The ‘‘Telugu Script Recognition—a Feature Based Approach’’ has been proposed by Purkaystha et al. [23]. The idea of Telugu characters being made up of circular pieces with varying radii is incorporated into the piece. In order to recognise a character, it has to break down into its component parts and figure out what each one does. Telugu

characters’ canonical shapes are preserved by using circular segments as the feature set. According to the research, recognition rates ranged from 78 to 90 percent across various subjects, and from 91 to 95 percent when the reference and test sets came from the same field. Recognising Telugu characters with Neural Networks is something that Maitra et al. [24] proposed. For the time being, the recognition process will use the Hopfield model of a neural network acting as an associative memory. Eventually, they advocated using multiple neural networks for associative memory because the Hopfield neural network had a storage capacity limit. These networks are based on training patterns that are not related to each other [25]. They were able to show that this plan may alleviate the storage issue.

3. Proposed Model

Character recognition on images is mostly used for character classification and recognition. Using Optical Character Recognition or intelligent word recognition can be used to perceive the text [26]. Recognition software takes care of formatting, character segmentation, and word detection. In this case, the pictures are all in grey scale that is converted from normal stage. Each image has a 52×52 pixel resolution. The image is shrunk to the following sizes during the preparation stage: 224×224 . Figure 3 shows the proposed model framework and procedure in image segmentation and pixel extraction.

The pooling layers are considered for accurate pixel extraction in which hidden layers are used for the process of feature extraction [27]. The character recognition process using segmentation and pixel extraction using CNN model with pooling layer representation is shown in Figure 4.

There are 4 fully connected layers and 12 hidden layers. One convolution layer and one pooling layer are present in each hidden layer. An image is considered having Telugu text as input and the image set is represented as $\text{Image_Set} = \{I_1, I_2, \dots, I_n\}$. The considered image is segmented into subimages as portions for accurate pixel extraction. The initial image will undergo a segmentation process as follows:

$$\text{Init_Seg}(\text{Image_Set}(M)) = \frac{\sum_{i=1}^P \text{Image_Set}(M)_i + T_h}{(p+1)} + \theta,$$

$$\text{Seg}(I(x, y)) = \sqrt{\frac{\sum_{i=1}^M (\text{Image_Set}_i - \min(\text{Init_Seg}))^2 + \min(\text{pixel}(i))^N}{\max(\text{grey_range}(\text{Pixel}(i))) + \lambda}} + \sqrt{\frac{\sum_{i=1}^M \min \text{range}(\text{Image_Set}(N))^2 + \max(\text{Init_Seg}) + \max(\text{pixel} + i)^2}{\text{grey_range}(\text{Pixel}(i))}}. \quad (1)$$

Here, M represents total images in the Image_Set , x, y are the pixel coordinate values, N is the total iterations to be performed, pixel is the current pixel and $\text{pixel} + i$ is the neighbour pixel range. Initially from the image set a single

image Image_Set_i is considered and then the grey levels of the image are calculated. The image pixels are extracted from each coordinate with x, y values [28]. The image is segmented into λ parts based on the size of the image. The pixel

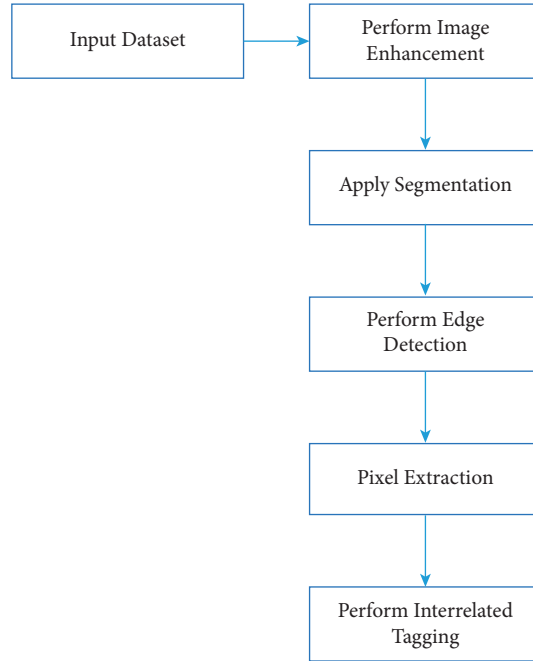


FIGURE 3: Shows the basic steps in the segmentation algorithm.

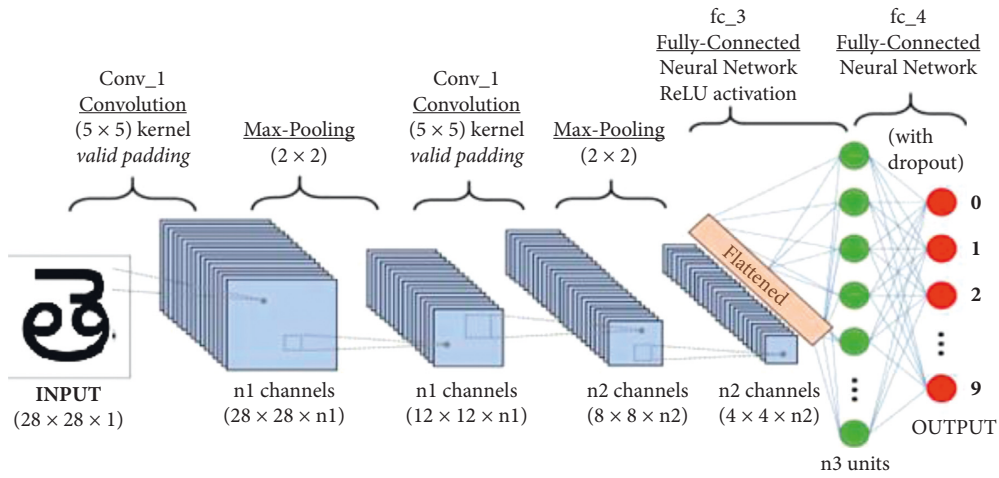


FIGURE 4: Pooling layer representation.

(i) represents a pixel in the segment extracted from coordinates x, y . The pixel having max intensity values is considered in the extraction process. T_h is the threshold range in which the grey range is considered. θ is the angle of the image provided as input.

The process of image segmentation divides the images into subimages and then filtering is applied on each segment to remove the noisy values. The filtering procedure is applied on the segments as follows:

$$\begin{aligned}
 \text{Filter}(\text{Seg}(\text{Image_Set}(M))) &= \text{filter}\left(\sum_{i \in M} x(\text{pixel}(i) + \text{pixel}(i + 1) + Th)\right), \\
 \text{Seg_Filter}(\text{Seg}(N)) &= \frac{\text{grey_range}(\text{Image_Set}(M))_{\text{avg}} + \max(\sum_{\text{pixel}(i)} \max(\text{grey_range}(\text{Seg}(i))))}{\max(\text{Filter}(\text{Seg}(M)))}.
 \end{aligned} \tag{2}$$

TABLE 1: Representation of pixel values from grey scale to binary.

<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr><td>21</td><td>34</td><td>55</td></tr> <tr><td>19</td><td>47</td><td>65</td></tr> <tr><td>87</td><td>23</td><td>145</td></tr> </table>	21	34	55	19	47	65	87	23	145	→	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td style="background-color: grey;"></td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	1	1	1	1		1	1	1	1	●	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr><td>2^0</td><td>2^1</td><td>2^2</td></tr> <tr><td>2^3</td><td style="background-color: grey;"></td><td>2^4</td></tr> <tr><td>2^5</td><td>2^6</td><td>2^7</td></tr> </table>	2^0	2^1	2^2	2^3		2^4	2^5	2^6	2^7
21	34	55																													
19	47	65																													
87	23	145																													
1	1	1																													
1		1																													
1	1	1																													
2^0	2^1	2^2																													
2^3		2^4																													
2^5	2^6	2^7																													
<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr><td>33</td><td>67</td><td>14</td></tr> <tr><td>56</td><td>29</td><td>24</td></tr> <tr><td>39</td><td>21</td><td>16</td></tr> </table>	33	67	14	56	29	24	39	21	16	→	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td style="background-color: grey;"></td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	1	0	1	0		1	1	1	0	●	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr><td>2^0</td><td>2^1</td><td>2^2</td></tr> <tr><td>2^3</td><td style="background-color: grey;"></td><td>2^4</td></tr> <tr><td>2^5</td><td>2^6</td><td>2^7</td></tr> </table>	2^0	2^1	2^2	2^3		2^4	2^5	2^6	2^7
33	67	14																													
56	29	24																													
39	21	16																													
1	0	1																													
0		1																													
1	1	0																													
2^0	2^1	2^2																													
2^3		2^4																													
2^5	2^6	2^7																													

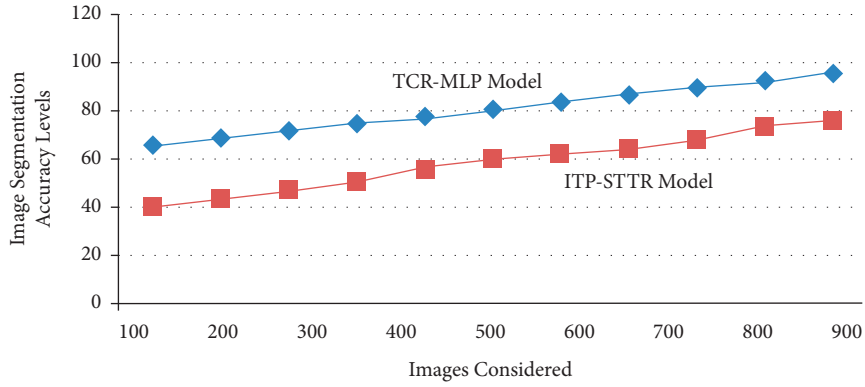


FIGURE 5: Image segmentation accuracy levels.

To perform filtering each pixel intensity values are considered by calculating grey level range of each pixel and the filtering is applied by considering the max range and min range and the min range values are removed to eliminate the noise from the segment of the image. pixel (i) and pixel ($i + 1$) are the pixel and its neighbouring pixel where the individual ranges are considered and filtered accordingly [29].

The edge detection procedure is applied on the filtered image segments so that the character shape and size are detected for accurate text pixel recognition. The edge detection process is performed as follows:

$$\text{Edge_Set}_{(\text{Seg-Filterer}(N))} = \frac{\text{pixel}_{\text{range}(i)\text{avg}} + \alpha}{\max(\sum_i \text{Min}(\text{Seg_Filter}(N(i))))} + \sum_{i=1}^M \left\{ \frac{\|\text{min}(\text{pixel}(\text{con}))\|}{\text{size}(\text{Image} - \text{Set}(m))} \right\}^N. \quad (3)$$

Here, α is the image default mean contrast levels considered in the process of filtering. The edge detection of an image is performed by comparing all pixel values to their neighbour pixel set and the difference in the pixel contrast is considered and the border between relevant and irrelevant pixels and the border and edges are finalised [30]. The

irrelevant pixels are considered as noisy values and eliminated and the pixels within the border and edge are extracted for character detection.

The pixel values are extracted from the edges detected and the pixels in the range of the edges are extracted as follows:

$$\text{Pixel_Set}(\text{Edge_Set}(N)) = \sum_N \text{Seg_Filter}_{i,j} + \text{grey}_{\text{range}(\text{Image}_{\text{set}})} + Th_N. \quad (4)$$

Each pixel grey value is extracted and maintained in the array set for further analysis. A character which is having a structure can be recognised only by considering the grey range and a pixel out of that range is not considered [31]. The process is repeated for all the pixels within the border and then such relevant and identified pixel in a character are marked as calculated grey value in the array. Table 1 shows the representation of pixel values from grey scale to binary.

If the grey value of the neighbour pixel pixelrange between min and max range, they are labelled as 0 or 1. The representation is stored in the array as.

The normalisation procedure is performed and the pixel intensity values are applied. The similarity difference of the pixels is analysed for text recognition in the image. The similarity difference is identified as follows:

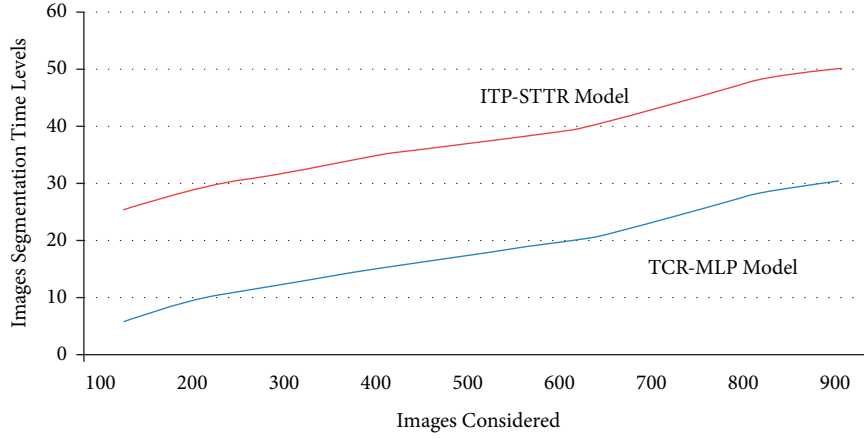


FIGURE 6: Image segmentation time levels.

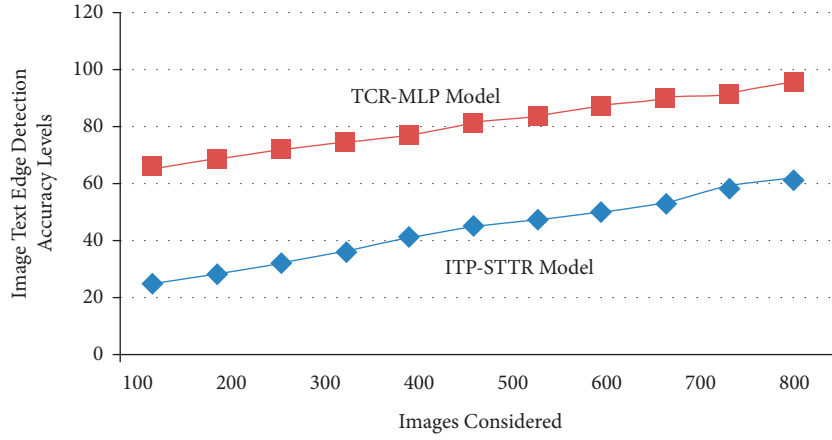


FIGURE 7: Image text edge detection accuracy levels.

$$\begin{aligned}
 & \text{Similarity}_{i,j}(\text{Image_Set}(N)) \\
 &= \sum_{i=1}^M \text{Edge}_{\text{Set}(\max(\text{filter}))} + \text{pixel}(i, i + 1) \\
 &+ \frac{\text{pixel}_{\text{range}(i)\text{avg}}}{\min(\sum_i \max(\text{Seg_Filter}(N(i))))}. \tag{5}
 \end{aligned}$$

The similarity difference of the pixels inside the edges and borders is need to be verified to form the character and the interlinking is performed based on the similarity levels. The pixel ranges are considered inside the border using pixelrange between pixel (i) and pixel ($i + 1$) inside the edges. The min value pixels are eliminated and the remaining are considered for character formation.

The error rate is measured as follows:

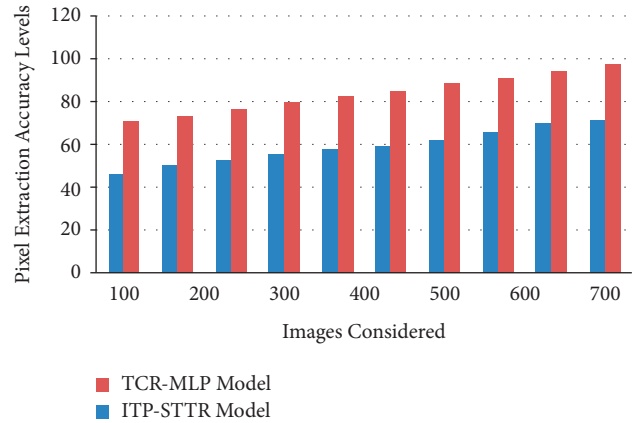


FIGURE 8: Pixel extraction accuracy levels.

$$\text{Mean Square Error Rate (Er)} = \frac{\sum_{x=1}^{M_n} \sum_{y=1}^N \left(\text{Pixel}_{\text{Set}}^{Th}(x, y) - \text{Seg_Filter}(x, y) \right)^2}{\text{size}(\text{Image}_{\text{Set}})}. \tag{6}$$

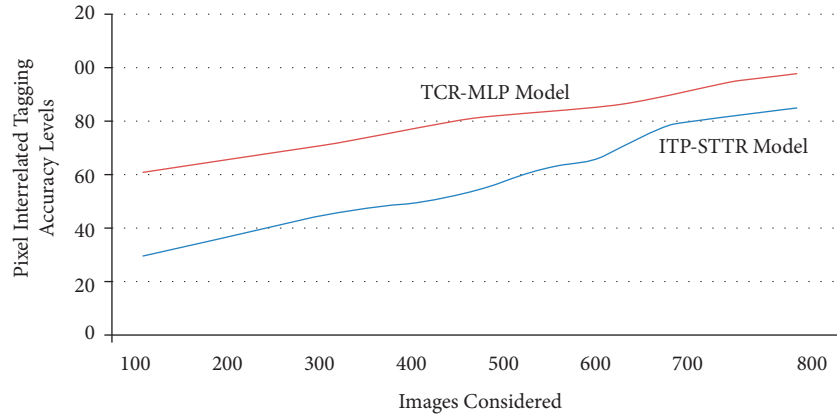


FIGURE 9: Pixel interrelated tagging accuracy levels.

4. Results

Research on online and offline recognition of Telugu handwriting characters has dwindled over the last few decades, making way for more notable and difficult computer vision applications, pattern recognition, and so on. In the domains of Image Analysis, Pattern Classification, and Computer Vision, recognition of Telugu character recognition is at an interesting juncture. Deep convolutional neural networks can now recognise characters, accompanying in a new era in character recognition. There are a variety of methods for recognising Telugu characters based on their handcrafted qualities. Performance is assessed by counting the segmented entities' matches to the ground truth's entities. Google Colab is used to assess the experimental findings implemented in python. Using the Telugu Characters dataset image as input, the proposed Interrelated Tagging Prototype with Segmentation for Telugu Text Recognition model is used for text recognition.

The proposed Interrelated Tagging Prototype with Segmentation for Telugu Text Recognition (ITP-STTR) Model is contrasted with Telugu Character Recognition using Multi-Layer Perceptron (TCR-MLP). The experiment on the suggested model was run in the Google Colab cloud environment with high-end specifications including 48 GB RAM, 128 GB ROM, and GPU Hardware Accelerator. The parameters considered are evaluated among the proposed and traditional models.

Image segmentation is a field of digital picture processing that deals with the division of an image into distinct portions based on their characteristics and attributes. Picture segmentation is the process of dividing an image into distinct regions with comparable characteristics. These spectrally comparable image segments are defined through the process of segmentation. The purpose of the segmentation process is to alter the image's properties into more relevant ones, making it easier to understand and categorise. The image segmentation accuracy levels of the proposed and traditional models are shown in Figure 5.

The segmentation process divides an image considered into multiple parts for thorough analysis. The proposed model in a less time completes the process of segmentation

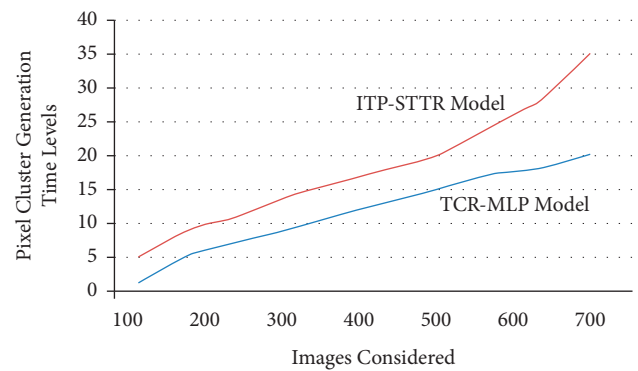


FIGURE 10: Pixel cluster generation time levels.

of images for better analysis. The segmentation time levels of the proposed and traditional models are shown in Figure 6.

The image text is detected after identifying the edges in the image. The image text detection accuracy levels are shown in Figure 7.

In digital imaging, the smallest piece of info in an image is pixels. Squares are used to represent pixels in a two-dimensional grid. As more images are sampled, they tend to be increasingly accurate representations of what the original image actually looks like. The pixel extraction accuracy levels of the proposed and traditional models are shown in Figure 8.

In the proposed research work, identification of a Telugu character is a critical process as it contains parts like *vattulu*, etc. To link this as a single character, pixel interrelated tagging is performed. The pixel interrelated tagging accuracy levels of the existing and proposed models are shown in Figure 9.

Clustering is a technique for locating groups of data that are similar to one another in a dataset. It is among the most widely used approaches in the field of data science. In each group, entities in that group are substantially more similar to beings in the other groups than they are to entities in the other groups. The pixels are clustered as sets for analysis of characters and the pixel cluster generation time levels of the proposed and traditional models are shown in Figure 10.

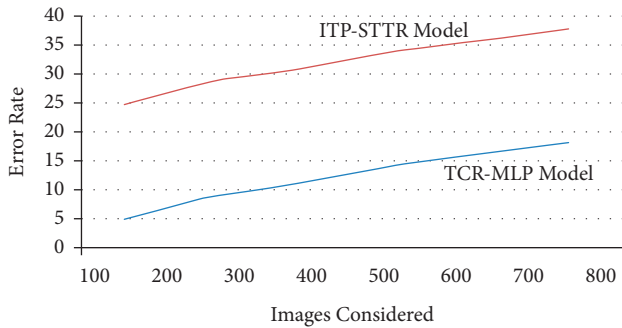


FIGURE 11: Comparison of error rate in TCR-MLP model and ITP-STTR model.

The error rate represents the failures in the identification of characters. The proposed and traditional model error rates are shown in Figure 11.

5. Conclusion

OCR for Telugu is currently a hot topic of study due to the wide range of potential applications. OCR for printed text continues to improve, although there is still room for improvement in the processing stage, character segmentation, and handling of broken letters. OCR for text recognition in images is extremely difficult, and the recognition rates are extremely low. No attempt has been made to fully recognise image text because there is no dataset for Telugu words available. There are many different characters in different languages, which makes handwritten character identification a difficult challenge. Instead, a variety of deep learning techniques are analysed to help us crack the code of character identification that is having less accuracy levels. In the proposed research, an efficient deep learning model with Interrelated Tagging Prototype with Segmentation for Telugu Text Recognition is introduced that accurately segments the image for accurate pixel extraction from the input image. The Telugu character can be found on real data by utilising OCR technology. Different character images have been selected in such a way that Telugu character images and training data are covered by the majority of classes and accomplished with a symbol accuracy rate. The image processing techniques are applied in the proposed model that involves image segmentation, edge and border detection, text pixel extraction to identify the text from the images and the pixel tagging are performed, and interrelated pixels are linked to form Telugu characters that are identified in an image. When compared to current state-of-the-art algorithms, the suggested method's results are rather impressive. Even on a larger dataset, the approach can enhance recognition accuracy and be extended to additional classes. The suggested CNN architecture in this work is trained exclusively for printed Telugu characters; however, this can be improved in the future to include handwritten Telugu characters as well. Consideration must be given to handwritten characters while developing a different segmentation approach.

Data Availability

The data used to support the findings of this study are included within the article. Should further data or information be required, these are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors thank GITAM Deemed-to-be-University for providing support to complete this research work.

References

- [1] M. S. Velpuru, P. Chatterjee, G. Tejasree, S. M Ravi Kumar, and R. Nageswara, "Comprehensive study of deep learning-based Telugu OCR," in *Proceedings of the IEEE ICSSIT 2020*, IEEE, Tirunelveli, India, August 2020.
- [2] V. V. Neerugatti, Manjunathachari, and Satyaprasad, "Handwritten Telugu composite character recognition using morphological analysis," *International Journal of Pure and Applied Mathematics*, vol. 119, pp. 667–676, 2018.
- [3] C. Bhagvati, S. Tanuku Ravi, K. Mahesh, and N. Atul, "On developing high accuracy OCR systems for Telugu and other Indian scripts," in *Proceedings of the Language Engineering Conference*, pp. 0–7695, IEEE, Hyderabad, India, December 2002.
- [4] N. S. Panyam, T. R. Vijaya Lakshmi, N. V. Koteswara Rao, T. V. Rajinikanth, and W. Abdul, "Telugu Handwritten Character Recognition Using Zoning Features," in *Proceedings of the 2014 International Conference on IT Convergence and Security (ICITCS)*, p. 978, IEEE, Beijing, China, October 2014.
- [5] K. C. Prakash, Y. M. Srikar, G. Trishal, S. Mandal, and S. S. Channappayya, "Optical Character Recognition (OCR) for Telugu: Database, Algorithm, and Application," in *Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 978–981, IEEE, Athens, Greece, October 2018.
- [6] K. Nagarajan, A. Rajagopalan, S. Angalaeswari, L. Natrayan, and W. D. Mammo, "Combined economic emission dispatch of microgrid with the incorporation of renewable energy sources using improved mayfly optimization algorithm," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 6461690, 22 pages, 2022.
- [7] A. sendrayaperumal, S. Mahapatra, S. S. Parida et al., "Energy auditing for efficient planning and implementation in commercial and residential buildings," *Advances in Civil Engineering*, vol. 2021, Article ID 1908568, 10 pages, 2021.
- [8] R. Achanta, H. Trevor, C. V. Chakradhar, B. Rajesh, and M. Raghavendra Reddy, "A Study on online Handwritten Telugu Character Recognition," *International Journal of Science, Engineering and Technology Research (IJSETR)*, vol. 5, no. 7, July 2016.
- [9] N. Shobha Rani and T. Vasudev, "Post-processing Methodology for Word Level Telugu Character Recognition Systems Using Unicode Approximation Models," in *Proceedings of the 2015 International Conference on Trends in Automation*,

- Communications and Computing Technology (I-TACT-15)*, pp. 978–981, IEEE, Bangalore, India, December 2015.
- [10] Magesh, “Pervasive computing in the context of COVID-19 prediction with AI-based algorithms,” *International Journal of Pervasive Computing and Communications*, vol. 16, no. 5, pp. 477–487, 2020.
- [11] D. K. Jain, S. K. S. Tyagi, S. Neelakandan, M. Prakash, and L. Natrayan, “Metaheuristic optimization-based resource allocation technique for cybertwin - driven 6G on IoE environment,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 7, pp. 4884–4892, 2022.
- [12] K. Karthick, K. B. Ravindrakumar, R. Francis, and S. Ilankannan, “Steps involved in text recognition and recent research in OCR,” *A Study International Journal of Recent Technology and Engineering*, vol. 8, pp. 2277–3878, May 2019.
- [13] C. Amit, “A review of various character segmentation techniques for cursive handwritten words recognition,” *International Journal of Information and Computation Technology*, vol. 4, 2014.
- [14] R. Kumar and A. Singh, “Character segmentation in gurmukhi handwritten text using hybrid approach,” *International Journal of Computer Theory and Engineering*, vol. 3, no. No. 4, pp. 499–501, 2011.
- [15] A. VenkataSrinivasa Rao, MadasuSubbarao, N. V. Rao, A. S. C. S. Sastry, and L. Pratap Reddy, “Segmentation of touching handwritten numerals and alphabets,” in *Proceedings of the 2009 Second International Conference on Computer and Electrical Engineering*, IEEE, Dubai, UAE, June 2009.
- [16] S. E. Benita Galaxy and S. Selvin Ebenezer, “Enhancement of segmentation and zoning to improve the accuracy of handwritten character recognition,” in *Proceedings of the International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, IEEE, Chennai, India, March 2016.
- [17] A. Darshan, N. Girdhar, R. Bhojwani et al., “Energy audit of a residential building to reduce energy cost and carbon footprint for sustainable development with renewable energy sources,” *Advances in Civil Engineering*, vol. 2022, pp. 1–10, Article ID 4400874, 2022.
- [18] S. S. Sundaram, “Smart clothes with bio-sensors for ECG monitoring,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 4, pp. 298–301, 2019.
- [19] A. Akm and K. T. Abdul, “Handwritten Arabic Numeral Recognition Using Deep Learning Neural Networks,” in *Proceedings of the 2017 IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, pp. 978–981, IEEE, Dhaka, Bangladesh, February 2017.
- [20] L. P. Natrayan, S. S. Sakthi, and J. Elumalai, “Analyzing the Uterine physiological with MMG Signals using SVM,” *International journal of pharmaceutical research*, vol. 11, no. 2, pp. 165–170, 2019.
- [21] M. Rajnoha, B. Radim, and M. K. Dutta, “Handwriting Comenia Script Recognition with Convolutional Neural Network,” in *Proceedings of the 2017 40th International Conference on Telecommunications and Signal Processing (TSP)*, pp. 978–981, IEEE, Barcelona, Spain, July 2017.
- [22] R. Zhang, Q. Wang, and Y. Lu, “Combination of ResNet and center loss based metric learning for handwritten Chinese character recognition,” in *Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition*, IEEE, Kyoto, Japan, November 2017.
- [23] B. Purkaystha, T. Datta, and Md S. Islam, “Bengali handwritten character recognition using deep convolutional neural network,” in *Proceedings of the International Conference of Computer and Information Technology (ICCIT)*, pp. 22–24, IEEE, Dhaka, Bangladesh, December 2017.
- [24] D. S. Maitra, U. Bhattacharya, and S. K. Parui, “CNN based common approach to handwritten character recognition of multiple scripts,” in *Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1021–1025, IEEE, Tunis, Tunisia, November 2015.
- [25] Asha, “IoT enabled environmental toxicology for air pollution monitoring using AI techniques,” *Environmental research*, vol. 205, Article ID 112574, 2022.
- [26] Anupama, “Deep learning with backtracking search optimization-based skin lesion diagnosis model,” *Computers, Materials & Continua*, vol. 70, no. 1, pp. 1297–1313, 2021.
- [27] S. Acharya, A. K. Pant, and P. K. Gyawali, “Deep learning based large scale handwritten Devanagari character recognition,” in *Proceedings of the 2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, pp. 1–6, IEEE, Kathmandu, Nepal, December 2015.
- [28] K. R. Vaishali, S. R. Rammohan, L. Natrayan, D. Usha, and V. R. Niveditha, “Guided container selection for data streaming through neural learning in cloud,” *International Journal of System Assurance Engineering and Management*, vol. 16, pp. 1–7, 2021.
- [29] S. D. Prasad and Y. Kanduri, “Telugu Handwritten Character Recognition Using Adaptive and Static Zoning Methods,” in *Proceedings of the 2016 IEEE Students’ Technology Symposium (TechSym)*, pp. 299–304, IEEE, Kharagpur, India, September 2016.
- [30] D. Singh, J. P. Saini, and D. S. Chauhan, “Hindi character recognition using RBF neural network and directional group feature extraction technique,” in *Proceedings of the 2015 International Conference on Cognitive Computing and Information Processing (CCIP)*, pp. 1–4, IEEE, Noida, India, March 2015.
- [31] J. Wang and S. Lee, “Data augmentation methods applying grayscale images for convolutional neural networks in machine vision,” *Applied Sciences*, vol. 11, no. 15, p. 6721, 2021.