A Unicode based Deep Handwritten Character Recognition model for Telugu to English Language Translation

***BV Subba Rao*[1] *J. Nageswara Rao*[2] *Bandi Vamsi*[3] *Venkata Nagaraju Thatha*[4] *Katta Subba Rao*[5]**

[1]Dept of Information Technology, PVP Siddhartha Institute of Technology
[2]*Department of Computer Science and Engineering, Lakireddy Bali Reddy College of Engineering, Mylavaram, NTR District, PIN- 521230, Andhra Pradesh, India,*
[3]*Department of Artificial Intelligence & Data Science, Madanapalle Institute of Technology & Science, Madanapalle - 517326, INDIA*
[4]*Department of Information Technology MLR INSTITUTE of technology Hyderabad 500049*
[5]*Department of Computer Science and Engineering, B V Raju Institute of Technology, Narsapur, Medak (District), Telangana, India*

Abstract Telugu language is considered as fourth most used language in India especially in the regions of Andhra Pradesh, Telangana, Karnataka etc. In international recognized countries also, Telugu is widely growing spoken language. This language comprises of different dependent and independent vowels, consonants and digits. In this aspect, the enhancement of Telugu Handwritten Character Recognition (HCR) has not been propagated. HCR is a neural network technique of converting a documented image to edited text one which can be used for many other applications. This reduces time and effort without starting over from the beginning every time. In this work, a Unicode based Handwritten Character Recognition(U-HCR) is developed for translating the handwritten Telugu characters into English language. With the use of Centre of Gravity (CG) in our model we can easily divide a compound character into individual character with the help of Unicode values. For training this model, we have used both online and offline Telugu character datasets. To extract the features in the scanned image we used convolutional neural network along with Machine Learning classifiers like Random Forest and Support Vector Machine. Stochastic Gradient Descent (SGD), Root Mean Square Propagation (RMS-P) and Adaptative Moment Estimation (ADAM)optimizers are used in this work to enhance the performance of U-HCR and to reduce the loss function value. This loss value reduction can be possible with optimizers by using CNN. In both online and offline datasets, proposed model showed promising results by maintaining the accuracies with 90.28% for SGD, 96.97% for RMS-P and 93.57% for ADAM respectively.

Keywords
Handwritten Character Recognition (HCR), Telugu Unicode's, Language Translation, ML optimizers

1. Introduction

Now-a-days the demand for automated models has drastically increased due to identification and extraction of characters that are available in image and documentation formats. The pixel quality of an image is one of the main factors to extend identification accuracy [1]. Hence, in order to overcome this limitation especially in image data like removal of noise, malformation in the image or picture at the time of scanning. Most of the language translation research works are carried out on globally recognized languages [2]. One of the most challenging tasks for recognition is Hand Written Character (HCR) particularly in English, Chinese and Arabic languages.So, it can be made clear that recognition of regional languages is a complex research area due to different shapes, size and stokes especially for the languages like Telugu in Andhra Pradesh and Telangana states of India [3].

India is considered as one of the most multilanguage country. There are 23 organization common languages in 12 different main scripts. Of them Telugu language is the ancient language having the consonants and vowel count of 16 and 32 respectively [4].Also, that characters that are designed through combining the consonants and vowels are 560 'guninthalu' having 612 letters. Due to different categories this becomes complex in identifying the Hand Written Telugu language letters particularly in 'guninthalu' [5]. Each letter is closely related to another letter in Telugu language. Many of the authors endeavoured in identifying the regional languages particularly Telugu language by matching the characters by the templates and other methodologies [6].

Manuscript received February 5, 2024
Manuscript revised February 20, 2024
https://doi.org/10.22937/IJCSNS.2024.24.2.12

Due to the increase in number of resources like offline written text, images, latest technology devices have advanced online Telugu data identification is considered as one of the active and challenging research areas in the study of pattern identification [7]. Telugu is considered as the regional language in Southern India and is the native language. For identification of the starting and ending letter, the Telugu language has many characters [8]. These are divided into two types consonant and vowel characters. In few languages it becomes more complex for recognition due to different shapes, size and the count of letters in particular language [9]. The Telugu language has more count of letters which represent the same shape [10]. The new technology of Handwritten Character Recognition (HCR) is used in making tasks easier. The main property of HCR is instead of using an image; this helps to convert mechanical translation into machine encoded text that can be of typed text, scanned paper, handwritten and a subtitle text superimposed [11]. To gather data from hard paper texts such as photos, journals and magazine involves in time consuming process [12]. To overcome this HCR is used to capture the image either from handwritten or typed text; also, it can capture the images through webcams [13]. By maintaining the accuracy of 90 percent, the typed text is captured and with an accuracy of 40-50 percent the handwritten text is read. This is very useful in converting the photos or scanned document into text that which can be accessed by the computer. It is mainly helpful for the data that is handwritten [14]. It is mainly used in the receipts related to banks, business cards, old documents and other relevant text documents. Developing technology helps in finding the innovative ways of finding all documents relying on knowledge with these HCR applications [15]. The HCRs are adopting extensive measures of the reports either in the form of typed formats aligned or handwritten to computer understanding code, without any change, variation and different parameters [16]. The handwritten analysis is normally referred as on-line and off-line identifications.

Recognition of the characters is enhanced through handwritten characters. On the other hand, automatic translators are required in case of offline handwritings. In this type, the text is converted to a picture of codes which can be used for analysis of text [17]. In general, offline handwriting is complex since there are many modes of handwriting from different individuals. The technique of HCR involves in analysing of the patterns, Artificial Intelligence, processing of the signal and machine vision [18]. These methodologies are commonly used in identification of the offline handwritten characters. They are used to identify the image of characters which is helpful to understand by the system [19]. This can be implicated with the use of mechanical or electronic translators without having any variations for pictures of the offline handwritten characters or data that is typed as text [20]. The working flow of the U-HCRmodelas shown in Figure 1.

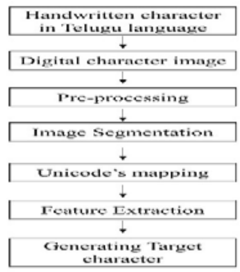


Figure 1: Proposed flow of U-HCR

In the field of video and image evolutions, internet is playing a key role in overflow of pictures as well as videos. These are giving enough ways for developing more research programs in the both representations to make understand better for common people regarding the complex tasks and methodologies [21]. Due to the updated technologies like Machine learning and Artificial Neural Networks (ANN) latest methodologies are evolved in the recent years. Of them the Convolution Neural Network (CNN) is one of the most innovative frameworks. By using this CNN in processing of an image is advantageous as ANN performance degenerated in identification of object and classification of an image. The better the CNN is accessible, the research with CNN in picture initialization of the attributes increased tremendously. These have various benefits in different categories including natural language processing (NLP), speech and vision identification.

The main objectives of this work are detailed below:

- The Telugu language characters / letters / digits in the scanned image are identified.
- During the pre-processing phase, misclassified letters and unwanted pixels are removed.
- Segmentation is applied on every letter/character in the input image.
- Based on the dimensions and shapes, the pattern is identified based on labelled images.
- According to the pattern, the Unicode's mapping is done on each character.
- Based on these Unicode's the concatenation among the letters becomes easier.
- Generating a Target language character by using feature extraction process.

The remaining sections are arranged as: Existing works and its limitations are discussed in 'Section-2: Related Works', Methodology of the proposed Unicode's based HCR is elaborated in 'Section-3: Methodology', The results and discussions of the proposed model with various parameters are discussed in 'Section 4: Results and Discussions'. The conclusion of this work with path to future scope is discussed in 'Section 5: Conclusion and Future Scope'.

2. Related Works

B. Soujanya et al. [22], In comparison to certain other languages such as English, Chinese and Japanese, the handwritten identification in Indian languages like Telugu, Hindi etc has received less emphasis. In this work, the authors used various optimization methods like gradient descent, Root Mean Square (RMS) and assessment methods like Adaptive estimation. These methods are used for classifying Telugu language character through convolution model. CNN can be used to get around the drawbacks of conventional methods for machine learning. Various handmade 'Telugu guninthalu' served as the input for the creation of the data set that served as the basis for this model. In comparison, the RMS optimizer performs 94.26% better than the Adaptive and gradient descentestimators.In order to recognize the spatial qualities of a character image, such strong filters frequently use weights learning from spatial characteristics of each lower level. Hyper Parameters are the values used in this model. When learning initially starts, these characteristics may have an impact on both the network design and the training.

N. Sarika et al. [23], The Convolutional model is working according by first detecting Telugu text as image letter by letter, analysing the image file, and then transforming the input image into ASCII codes.The OCR technology is used to convert text contained in an image into textual form. Pre-processing, character segmentation, features extraction, and postprocessing are the three primary components of the OCR technique.This recognition model produces an accuracy with 92% and it was trained using a Telugu characters large dataset that can contain up to 1600 characters.

R. Kibria et al. [24], There are numerous important possibilities for a systematic method of handwritten character recognition and identification using SVM. This method extracts characteristics to categorise the deep structural characteristics of Bangla Language characters. The feature extracted classification methods like Histogram gradients, Diagonal, Partitioning was used. Each letter is given a length of 80 variable feature representation as an outcome of this strengthened, potent combination of characteristics which is proven to be sufficient to accurately and individually identify every letter.The collected vectors are then employed in the SVM classifier for training and testing phases, when characters are classified and evaluated to their assigned categories. The findings obtained demonstrate greater classifier performance accuracy with 88.73%.

Vijaya Krishna et al. [25], Telugu text detection system converts printed and handwriting characters into appropriate text formats, is still a challenging issue. To achieve this, in this study a powerful Telugu Language text identification model for both printed and hand-written symbols is introduced. This method seeks to find and identify both printed and handwritten letters that are present in an input image. Mainly, the adaptive filtering approach is used to pre-process an image.To create useful regions, techniques for line, letter, and text separation methods are used. Additionally, feature extraction uses the combination of Efficient and Capsule Networks.In order to find and identify different characters that are present within the same image, a detection technique is developed. This technique includes a variety of functions, including pre-processing, character segmentation, extraction of features relying on optimizer, fusion and hyperparameters.

Ramegowda et al. [26], The development of a convolutional neural model for automated character segmentation. The written images are first obtained from the Chars74K and MADbase numeral datasets, followed by the data is pre-processed using skew identification and Gaussian filtration methods.Furthermore, the projected characteristic and thresholding methods are used to separate the specific lines and letters from the input images.The characteristics are used to features extracted from the segmentation results, and the Elephant Herding Optimization (EHO) approach is used to choose the discriminant features.The memory-based network classifies various classes such as 64 in English, 10 in Arabic, and 657 in Kannada using the chosen features.On the 'chars74K' and 'MADbase' digits datasets, this proposed model recognisedArabic, Kannada and English with 99.93%, 96.67% and 96.66%accuracy. To enhance the quality of the handwriting images, gaussian filtering and skew identification approaches are used.

Srinivasa Rao et al. [27], When constructing compound characters, there have been a large number of possible combinations of modifiers and consonants. A composite character is created when several symbols are combined. Telugu has a large number of output groups, which leads to significant interclass variance.We present two methods to enhance character or glyph classification in a Telugu OCR system. The capacity to recognise Telugu language is essential for Telugu OCR.Network nodes in an image are groups of similar pixels that are joined together using either 4- or 8-pixel connection.This research develops a 95% effective deep learning approach with interconnected tagging prototype and classification for Telugu character recognition.

Srinivasa Rao et al. [27], The automatic model that is being suggested can recognise both handwriting Bangla and Odia digits and letters with optimum optimization performance. This suggested model primarily addresses the three components of the characteristic hyperparameter.First, multidimensional characteristics from the character images are derived using the discrete transform. Next, the features

3. Methodology

3.1 Hand Written Recognition (HCR) with Unicode's

The HCR methodology consists of various phases namely, digitization, segmentation, pre-processing and extraction of the features are as follows:

- **Digital character:** The process of converting hand written data into machine understandable language is referred to as digitization. In this method each character contains of each data. The document that is scanned using this method is electrically transferred and showed in a picture format. Under this process, the digital picture is scanned through various type of sources are considered as future phase in the initialization stage.
- **Pre-Processing:** In this phase the group of operations is performed on the scanned input picture. Due to this the rendering of the picture is increased such that the 'Gray' stage text picture is initialized to fit in size of window which can be used for segmentation phase. Once after the noise is removed, then the bitmap picture is developed. Later this picture is converted into concentrated picture.
- **Image Segmentation:** In this phase, the picture is removed from the initial characters. It becomes complicated to divide into different categories in case of hand written characters when compared to that of the normal printed document. This is because of the scanned characters size, number of words, kind of paragraph, its scale and slant. Sometimes it becomes possible to merge two adjacent letter which becomes complex in the segmentation phase. While an enhancement is done in the either of the zones this complication of overlapping arises.
- **Unicode's mapping:**It is defined as the standard encoding system that is used to define the characters or letters from different languages. Likewise in case of Natural Language Processing (NLP), it handles various languages with numerous character groups. Each letter is encoded with a different integer point of code that lies among 0 and 0*10FFFF. It can be a sequence of zero or many points of code. During the encoding formats if the Unicode value exceeds, it utilizes one byte for every letter and is technically referred as point of code. The main advantage of these Unicode's is that it can be used in any type of system without any loss of data.
- **Extraction of Features:** During this phase, the isolated letters are separated. These attributes of the input letter make it work to note a letter in such a way. In order to determine the letters or characters different alignments are used. Few of them are Crossover, convert letters, open edges, diagonal, letters, area letters, directional factors, parabolic curve shaped factors and power shaped groups.

The structure of Telugu language is mainly depending as Vowels → Dependent vowels → Consonants. The Unicode values for Telugu vowels as shown in Figure 2. The dependent vowel signs along with digits and its Unicode's are shown in Figure 3. The list of Unicode values for Telugu consonants are given in Figure 4.

Telugu letter	అ	ఆ	ఇ	ఈ	ఉ	ఊ	ఎ	ఏ	ఐ	ఒ	ఓ	అం	అః
Similar English letter(s)	a	aa, AA, aA, Aa	i	ee, cE, EE, Ee	u	uo, Uo, Oo, UO	E	ae, E, Ae, AE	ai, AI, aI	o	oa, O	aM, AM, am, Am	Aha, AHA
Unicode's	U+0C 05	U+0 C06	U+0 C07	U+0 C08	U+0 C09	U+0 C0A	U+0 C0E	U+0 C0F	U+0 C10	U+0 C12	U+0 C13	(U+0C05) + (U+C0C02)	(U+0C06) + (U+0C03)

Figure 2: List of Telugu language vowels with Unicode's

Telugu sign	ా	ి	ీ	ు	ూ	ృ	ౄ	ె	ే	ై	ొ	ో	ౌ
Unicode's	U+0 C3E	U+0 C3F	U+0 C40	U+0 C41	U+0 C42	U+0 C43	U+0 C44	U+0 C46	U+0 C47	U+0 C48	U+0 C4A	U+0 C4B	U+0 C4C
List of Digits in Telugu language													
Telugu Digit	౦	౧	౨	౩	౪	౫	౬	౭	౮	౯			
Number	0	1	2	3	4	5	6	7	8	9			
Unicode's	U+0 C66	U+0 C67	U+0 C68	U+0 C69	U+0 C6 A	U+0 C6B	U+0 C6 C	U+0 C6D	U+0 C6E	U+0 C6F			

Figure 3: List of dependent vowel signs, digits and its Unicode's

Telugu letter	క	ఖ	గ	ఘ	ఙ	చ	ఛ	జ	ఝ	ఞ	ట	ఠ	డ
Similar English letter(s)	Ka	Kha	ga	gha	Gna	cha	Cha	ja	jha	ini	Ta	Tha	Da
Unicode's	U+0C15	U+0C16	U+0C17	U+0C18	U+0C19	U+0C1A	U+0C1B	U+0C1C	U+0C1D	U+0C1E	U+0C1F	U+0C20	U+0C21
Telugu letter	ఢ	ణ	త	థ	ద	ధ	న	ప	ఫ	బ	భ	మ	య
Similar English letter(s)	Dha	Na	ta	tha	da	dha	na	pa	pha	ba	bha	ma	ya
Unicode's	U+0C22	U+0C23	U+0C24	U+0C25	U+0C26	U+0C27	U+0C28	U+0C29	U+0C2B	U+0C2C	U+0C2D	U+0C2E	U+0C2F
Telugu letter	ర	ల	వ	శ	ష	స	హ	ళ	ఱ				
Similar English letter(s)	ra	la	va	sa	Sa	sha	ha	La	Ra				
Unicode's	U+0C30	U+0C32	U+0C35	U+0C36	U+0C37	U+0C38	U+0C39	U+0C33	U+0C31				

Figure 4: List of Telugu language consonant with Unicode's

3.2 Centre of Gravity partitioning

To recognize letters in an image, the Centre of Gravity (CG) is used in feature extraction phase. The characters gravity centre format is initially used in identifying its location, which position the data of character like size, shape, dimension etc. are obtained. Now the raw image is analysed by removing its edges.The area around the character is scanned on that edge image to identify particular size of the character. The positions of $CG(C_x, C_y)$ for any character structure are given by equations 1 and 2. The centre identification by CG approach is shown in Figure 5.

$$C_x = \frac{1}{m*n} x * f(x,y) \; ; \; C_y = \frac{1}{m*n} y * f(x,y) \tag{1}$$

$$f(x,y) = \begin{Bmatrix} 1; \; \forall \; black \; pixels \\ 0; \; Otherwise \end{Bmatrix} \tag{2}$$

Here, $m * n$ represents the dimensions or number of pixels in an image.

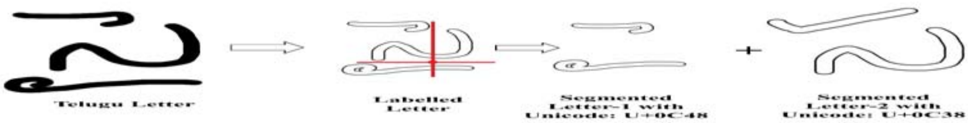


Figure 5: Centre based partitioning of a character using CG

To obtain the feature vectors the highlighted black column for every character based on its row, column and diagonals are required.These feature vectorsF_v are generated based on following steps.

Step 1: The centre identification of a character is well recognized on labelled image based on top, bottom- and base-line strokes in input image.

Step 2: The pixel in the concerned direction is labelled and the preceding pixel in the same direction is also labelled then the count will be increased.

Step 3: After attaining the initial pixel in a group of raw pixels, the count is compared with the highest length of sequential labelled bars.

Step 4: Larger group of labelled bars for every character with particular direction is added and normalized to the total count of labelled pixels in the partitioned image is given by the equation 3.

$$F_v = \frac{Sum \; of \; highest \; pixels}{Total \; number \; of \; black \; colored \; pixels} \tag{3}$$

3.3 Convolution Neural Network (CNN)

The most commonly used model is CNN that has spread to tackle an enormous range of visual picture applications, categorization of the items and different formats of audio. These are brought into use by using the mathematical notations. In a multi-layer network, there will be multiple layers. This method takes raw pixel values as input rather than factor vectors that are involved in machine learning. There were different types of CNNs, since they have the same normal framework. The structures include a pooling layer, completely connected layer and a convolution layer as shown in Figure 6 and the summary of the CNN with 35,64,36 trained parameters is given in Table 2.The working procedure are as follows:

- **Convolution Layer:** In this layer the pictures entered as input are cleaned using this layer. This is used in recognizing the letters which are required to identify identical regions while analysing. The stretched picture needs the convolution method with the required attributes. Using a kernel, the input data is converted to a factor map through neural network.
- **Pooling Layer:** In this layer the features are gathered from the convolution layer are received. On maintain the crucial text, it also decreases the size of the picture. It obtains the best fir value by obtaining the larger value from each layer. In this layer it reduces the image to lower the count of attributes and probabilities in the process. One of the complex techniques in this is max pooling. It always chose the higher elements from the factor map covered through the filter.
- **Fully connected layer:**The extracted high-level image that is filtered are divided through various layer in the last layer. Every attribute present in the layer is connected to each other below it. In most of the designs the convolution and pooling are interconnected in many of the models.

Table 2: Summary of CNN architecture

Layer Name	Output Shape	Number of parameters
Convolution_2D_1	28x28x32	320
Leaky_ReLU activation		0
Max_pooling_1	14x14x32	0
Convolution_2D_2	14x14x64	18496
Leaky_ReLU activation		0
Max_pooling_2	7x7x64	0
Convolution_2D_3	7x7x128	73856
Leaky_ReLU activation		0
Max_pooling_3	4x4x128	0
Flatten Layer	2048	0
Dense Layer	128	262272
Leaky_ReLU activation	128	0
Dense Layer	10	1290
Total parameters		35,64,36
Total trained parameters		35,64,36

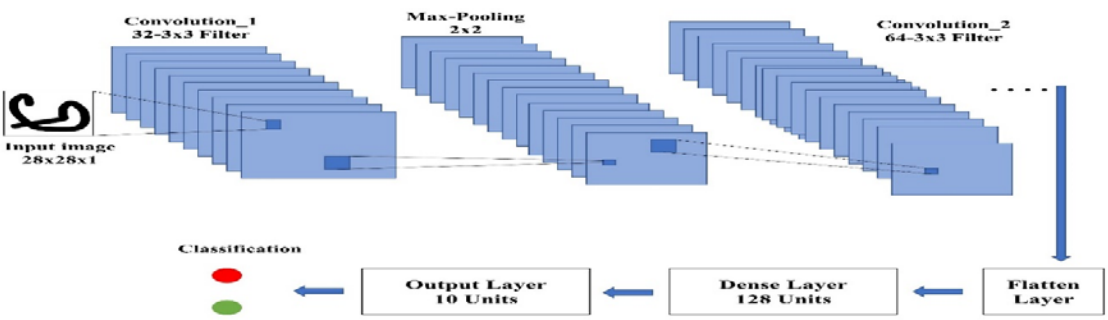


Figure 6: Structure of CNN architecture

From Figure 6, by using Flatten layer a one-dimensional vector is formed by conversion of two-dimensional matrix data before generating the Fully connected layers. Now, by activating 'ReLU' function we completely work on the adjacent layers. A normal regularised layers is a group to remove 20% of the vector randomly to avoid overfitting.

Finally, a sigmoid function is enabled to output layers having 10 neurons. A crucial region of CNN is its activation function that defines the output of a particular vector depending on a group of inputs. This function is utilised in establishing non-linearity to the proposed model. The performance of the CNN model is enhanced through choosing the necessary activation functions. The sigmoid and ReLU methods are used in this proposed model for obtaining proper classification

3.4 Machine Learning functions and optimizers

The performance of any machine learning models can be analysed though training, testing and validation process. After every cycle, the errors are determined using training and validation stages. The training phase is set to be completed when the count of epochs present in training and validation are not changed significantly. Whereason the test set errors in training and validation are identified and the network is estimated. This is a crucial stage in using the optimizers to enhance accuracy and reduce the errors. The optimisers help in changing the setting of weights for reducing the loss rate. To obtain the good performance model we include the group of limitations. They are: Rectified Linear Unit (ReLU), Sigmoid, Stochastic Gradient Descent (SGD), Root Mean Square Propagation (RMS-P) and Adaptive Moment Estimation (Adam).

3.4.1 ReLU

In this function, if the given input is negative, the ReLU function returns '0'. If the given input is positive then this function returns its value. This function is carried out very quickly than other functions. It is considered as default activation function for different kinds of neural networks. The mathematical equation for ReLU is given in equation 4.

$$f(x) = \max(0, x) \tag{4}$$

3.4.2 Sigmoid

It is given the equation 5, which represents the methodology in decision making from a range of 0 to 1. This function is used to predict the output which would be more accurate.

$$f(x) = \frac{1}{(1+e^{-x})} \tag{5}$$

3.4.3 Stochastic Gradient Descent (SGD)

It is a recursive process in enhancing the function of an object with required features. Since, it restores the original gradient by an estimated value. To attain quicker recursions for low convergence value, particularly in high dimensional enhancing issues it lowers the high computational load.

3.4.4 Root Mean Square Propagation (RMS-P)

It is equivalent to Momentum. The method to reduce the motion in the y axis region and increase the up-gradient pitch is given equation 6. Here, learning rate is denoted by 'n' which initial initialized value of 0.001, 'v_t' defines exponential squares of gradients and 'g_t' gradient value of time 't'.

$$\Delta\omega_t = -\frac{n}{\sqrt{v_t + \varepsilon}} g_t \tag{6}$$

3.4.5 Adaptive Moment estimation (ADAM)

This is considered as one of the important methods in Machine learning. It is a process to select the random probability pattern to enhance in solving linear systems having noisy data. It evaluates the adaptive learning rates for every character. Its iterative nature make itself in obtaining the near values of functions which only can be evaluated through noisy results can obtained by equation 7 and 8.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) * g_t \tag{7}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) * g_t \tag{8}$$

Here, 'g_t' gradient value of time 't', β_1, β_2 are gradient features, m_t and v_t are moving means. By default, the ADAM parameters having values of β_1 = 0.9 and β_2=0.999.

3.5 Dataset

Due to limited access to public data for training the HCR purpose, we created our own dataset.Various font characters are first gathered and scanned into accurate symbols. Table 3 and Table 4 shows the summary of online and offline Telugu HCR dataset after Augmentation technique. Figure 7 and 8 represents the scanned Telugu language characters for both online and offline.

Table 3: Summary of Online Telugu HCR dataset [28]

Serial No	Type	Number of Telugu characters	Total number of Telugu characters after Augmentation
1	Achulu	16	9,545
2	Guninthalu	34	20,590
3	Hallulu	36	23,891
4	Othulu	36	25,456

Table 4: Summary of Offline Telugu HCR dataset

Serial No	Type	Number of Telugu characters	Total number of Telugu characters after Augmentation
1	Achulu	16	10,591
2	Guninthalu	34	17,536
3	Hallulu	36	28,325
4	Othulu	36	23,567

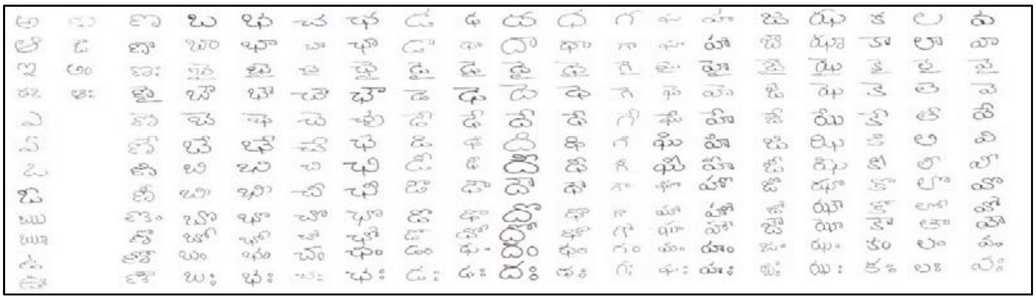


Figure 7: Online Telugu characters used in this work [28]

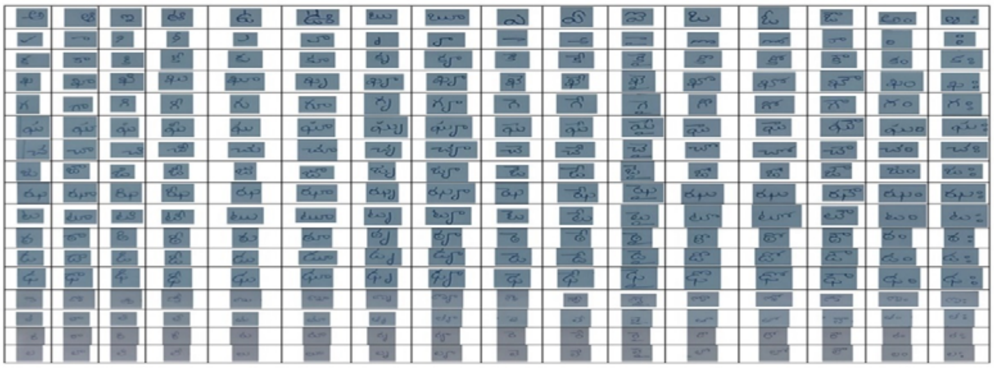


Figure 8: Offline Telugu characters created for this work

3.6Machine Learning Classifiers

In this work, we have used two ML classifiers which are independently tested and trained on both offline and online datasets. The classifiers which are used are Random Forest (RF) and Support Vector Machine (SVM).

3.6.1 Random Forest

This is used in classification and regression and is defined as an ensemble learning model. It is used in building the decision trees during training period and showing the class which is the mode of classes. These are the combo of tree identifiers in which every tree lies on the values independently of random vector sample having equal distribution of every tree in the region. The main factor is that the set of weak learners are gathered to generate the strong learner. Developing the necessary type of randomness make the classifiers more accurate.

It can be seen as a weighted neighbourhood. This is developed through training data set that make identifications $y$for every new value $x$by observing the surrounding of the point and is given by weight function 'W'.This can be represented in the form given in equation 9.

$$Y_{Predict} = \sum_{i=1}^{n} W(x_i, x')\, y_i \tag{9}$$

3.6.2 Support Vector Machine

Let us assume a binary classification problem having data set $(x_{1,}y_1) \ldots\ldots (x_{n,}y_n)$ where $x_{i,}$ represent input vector and $y_{i,}$ represent is a binary label for a kernel k. Solving the quadratic programming problem by a soft-margin support vector is represented by the equation 10.

$$k = y_1 * \alpha_1 + y_2 * \alpha_2 \tag{10}$$

3.7 Proposed Model

Figure 9, demonstrates the U-HCR model architecture of convolutional approach using for converting handwritten Telugu language to English language. Initially, the handwritten characters are scanned. The characters in the scanned image are identified individually according to available trained data. The data consists either single or concatenated characters. These can be segregated by CG. Each character that is separated represents a neuron based on deep neural network architecture. With the help of hidden layers, the identified character can be recognized as individual one or concatenated one. The mapping of input layer, hidden layer and output layer becomes more complex if we use only traditional neural networks. To map the input character and require output character we required convolutions for extracting the futures in it.

The feature extraction without number coding requires a greater number of epochs to train the available data and also need augmentation approaches. To improve the model accuracy and to reduce the loss parameters, in this work we used Unicode based number mapping for each Telugu language character. By using Unicode's, the structure of the model remains constant and number of mappings can be less. In Unicode mapping, there is separate unique numbers for vowels, consonants, digits and dependent vowel signs. Due these unique values separation and concatenation becomes easier because of continues numbers. For example,U+0C40, U+0C41, U+0C42 etc. These continues numbers can be helped in identify the misclassified characters for mapping proper character in Target language.

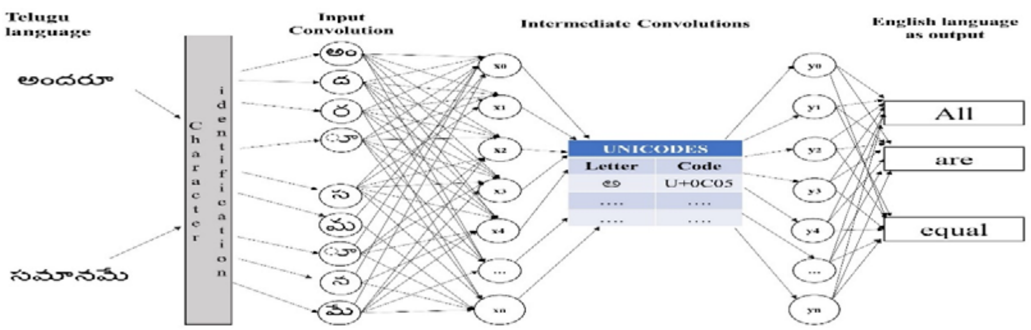


Figure 9: Proposed model

4. Results and Discussions

In HCR, the developed Unicode's based mapping model is implemented on both offline and online datasets. The performance of our U-HCR is evaluated by using ML based performance evaluation metrics like accuracy, precision, Recall and F1-score based on confusion matrix.

4.1 Performance evaluation metrics

The recall is calculated as the proportion of properly identified handwritten character images to all of the samples in the dataset is depicted by equation 11.The ratio of correctly classified handwritten character images to all the samples that have been categorised is known as the precision and is given by equation 12. Whereas equation 13 provides the mathematical formula, which states that the proportional mean of precision and recall is expressed as F1-score.

$$Recall = \frac{True\ positives}{True\ positives + False\ negative} * 100 \tag{11}$$

$$Precision = \frac{True\ positives}{True\ positives + False\ positive} * 100 \tag{12}$$

$$F1 - score = \frac{2 * True\ positives}{False\ positives + 2 * True\ positive + False\ negative} * 100 \tag{13}$$

The efficiency of our U-HCR model can be summed up by accuracy. It is determined by dividing the total number of estimated predictions by the number of accurate ones. The appreciation of accuracy is represented by mathematical equation 14.

$$Accuracy = \frac{True\ positives + True\ negative}{Total\ number\ of\ samples} \tag{14}$$

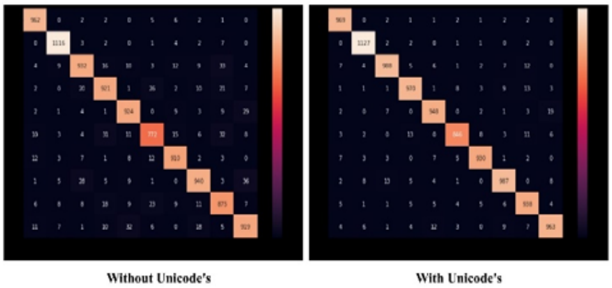


Figure 10: Confusion matrix with RMS-P optimizer

Table 5: Experimental results on offline dataset with Unicode's

Model	Optimizer	Accuracy	Precision	Recall	F1-score
CNN	SGD	88.95	81.08	84.39	86.34
	RMS-P	96.97	95.52	94.13	95.66
	ADAM	91.12	93.92	89.93	92.41
CNN-RF	SGD	90.21	89.93	87.19	89.59
	RMS-P	96.96	94.18	95.90	95.57
	ADAM	93.57	93.20	93.44	94.30
CNN-SVM	SGD	90.28	93.4	90.90	88.26
	RMS-P	93.99	95.50	95.82	93.54
	ADAM	92.02	91.92	91.02	90.84

Table 6: Experimental results on online dataset with Unicode's

Model	Optimizer	Accuracy	Precision	Recall	F1-score
CNN	SGD	88.14	80.36	83.87	85.91
	RMS-P	96.16	94.8	93.61	95.23
	ADAM	93.31	93.2	89.41	91.98
CNN-RF	SGD	89.4	89.21	86.67	89.16
	RMS-P	96.15	93.46	95.38	95.14
	ADAM	92.76	92.48	92.92	93.87
CNN-SVM	SGD	89.47	92.68	90.38	87.83
	RMS-P	93.18	94.78	95.3	93.11
	ADAM	91.21	91.2	90.5	90.41

Table 7: Experimental results on offline dataset without Unicode's

Model	Optimizer	Accuracy	Precision	Recall	F1-score
CNN	SGD	84.02	74.35	78.94	81.91
	RMS-P	92.04	88.79	88.68	91.23
	ADAM	89.19	87.19	84.48	87.98
CNN-RF	SGD	85.28	83.2	81.74	85.16
	RMS-P	92.03	87.45	90.45	91.14
	ADAM	88.64	86.47	87.99	89.87
CNN-SVM	SGD	85.35	86.67	85.45	83.83
	RMS-P	89.06	88.77	90.37	89.11
	ADAM	87.09	85.19	85.57	86.41

Table 8: Experimental results on online dataset without Unicode's

Model	Optimizer	Accuracy	Precision	Recall	F1-score
CNN	SGD	83.21	73.63	78.42	81.48
	RMS-P	91.23	88.07	88.16	90.8
	ADAM	88.38	86.47	83.96	87.55
CNN-RF	SGD	84.47	82.48	81.22	84.73
	RMS-P	91.22	86.73	89.93	90.71
	ADAM	87.83	85.75	87.47	89.44
CNN-SVM	SGD	84.54	85.95	84.93	83.4
	RMS-P	88.25	88.05	89.85	88.68
	ADAM	86.28	84.47	85.05	85.98

Table 9 and 10, represents the training accuracy and loss values on CNN model with and without Unicode's. This shows the improvement in training accuracy with respective to epochs for model fitting.

Table 9: Training results of CNN without Unicode's

Epoch Number	Training accuracy	Training loss	Time
1	67.30	68.45	13s 3ms/step
2	79.11	28.15	13s 3ms/step
3	80.57	23.15	13s 3ms/step
4	81.13	20.57	13s 3ms/step
5	81.47	19.46	13s 3ms/step

Table 10: Training results of CNN model with Unicode's

Epoch Number	Training accuracy	Training loss	Time
1	86.38	44.51	4s 2ms/step
2	96.95	10.24	4s 2ms/step
3	97.83	7.36	4s 2ms/step
4	98.31	5.42	4s 2ms/step
5	98.65	4.38	4s 2ms/step

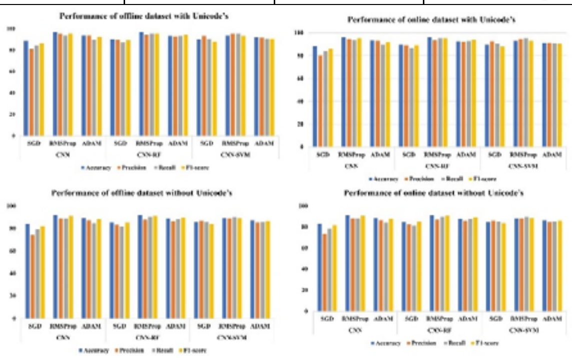
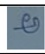
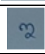
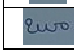
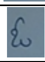
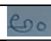


Figure 11: Visualization of experimental results

Table 11 represents the experimental results of accuracy all optimizers used in this work on Telugu vowel hand written characters. When compared to SGD, ADAM optimizers, the RMS-P has produced more accuracy values under training and testing phases.

Table 11: Experimental results of accuracyon Telugu vowels using Optimizers

Telugu Character	Training accuracy			Testing accuracy		
	SGD optimizer	RMS-P optimizer	ADAM optimizer	SGD optimizer	RMS-P optimizer	ADAM optimizer
	96.84	97.89	96.14	86.94	91.81	85.55
	96.31	97.91	96.44	87.13	91.86	85.7
	97.14	97.99	96.55	87.21	92.83	85.73
	96.29	97.95	84.55	87.01	92.31	85.63
	95.94	96.18	83.98	85.81	91.21	84.4
	95.76	96.09	82.98	84.71	90.02	82.96
	97.34	98.39	95.59	87.31	94.01	85.46
	95.64	95.29	93.94	85.53	90.21	83.96
	97.04	98.89	96.66	87.61	93.01	85.8
	95.14	96.79	93.86	85.53	90.21	83.78
	96.74	98.09	96.23	87.51	92.45	85.96
	97.29	98.15	96.46	87.94	92.86	86.12
	97.13	97.51	95.32	86.63	91.45	84.7
	95.89	97.6	95.36	86.72	91.54	84.88
	96.31	96.79	95.25	86.52	91.13	84.59
	96.89	97.09	95.95	86.64	91.23	84.69

5. Conclusion and Future Scope

HCR text identification particularly in image format is highly complex and also the identification rate is reliably low. Since then there is no methodology in completely identifying the characters in the image format for Telugu language due to lack of datasets and its high complex structure. The Telugu language comprises various characters in multiple languages that can be either hand written or scanned which are difficult for identification. To overcome this limitation, many techniques in deep learning are introduced to track the mapping between one language to another language with heavy training epochs. To improve the training accuracy and to minimize the number of epochs in training phase, a Unicode based HCR (U-HCR) is developed. This is used for mapping a scanned handwritten character from Telugu language to English language. To reduce the learning loss rate various optimizers are used such as SGD, RMS-P and ADAM on every type of Telugu character.In order to extract the features in scanned handwritten image character, our proposed model uses CNN architecture along with RF and SVM classifiers. These classifiers are used to classify the highest valued pixels given by CNN model. In future scope, there is a chance of converting this type of handwritten characters in a paragraph or bulk of pages at once using memory based deep learning models.

Reference

[1] Meena, B., Rao, K. V., &Chittineni, S. (2022). A Novel Method to Auto Configure Convolution Neural Network Model Using Soft Computing Technique to Recognize Telugu Hand-Written Character for Better Accuracy. Journal of Theoretical and Applied Information Technology, 100(18).

[2] Das, M. S., Reddy, C. R. K., Rahul, K., & Govardhan, A. (2011). Multilingual Optical Character Recognition System for Printed English and Telugu Base Characters. International Journal of Science and Advanced Technology (ISSN 2221-8386), 1(4), 106-111.

[3] Guptha, N. S., Balamurugan, V., Megharaj, G., Sattar, K. N. A., & Rose, J. D. (2022). Cross lingual handwritten character recognition using long short term memory network with aid of elephant herding optimization algorithm. Pattern Recognition Letters, 159, 16-22. https://doi.org/10.1016/j.patrec.2022.04.038

[4] Sonthi, V. K., Nagarajan, S., &Krishnaraj, N. (2022). An Intelligent Telugu Handwritten Character Recognition using Multi-Objective Mayfly Optimization with Deep Learning Based DenseNet Model. Transactions on Asian and Low-Resource Language Information Processing. https://doi.org/10.1145/3520439

[5] Shekar, K. C., Cross, M. A., & Vasudevan, V. (2021). Optical Character Recognition and Neural Machine Translation Using Deep Learning Techniques. In Innovations in Computer Science and Engineering (pp. 277-283). Springer, Singapore. https://doi.org/10.1007/978-981-33-4543-0_30

[6] Sethy, A., Patra, P. K., & Nayak, S. R. (2022). A Hybrid System for Handwritten Character Recognition with High Robustness. Traitement du Signal, 39(2). https://doi.org/10.18280/ts.390218

[7] Sharma, R., & Kaushik, B. (2022). Handwritten Indic scripts recognition using neuro-evolutionary adaptive PSO based convolutional neural networks. Sādhanā, 47(1), 1-19. https://doi.org/10.1007/s12046-021-01787-x

[8] Sankara Babu, B., Nalajala, S., Sarada, K., Muniraju Naidu, V., Yamsani, N., &Saikumar, K. (2022). Machine Learning Based Online Handwritten Telugu Letters Recognition for Different Domains. In A Fusion of Artificial Intelligence and Internet of Things for Emerging Cyber Systems (pp. 227-241). Springer.

[9] Ganji, T., Velpuru, M. S., &Dugyala, R. (2021). Multi variant handwritten telugu character recognition using transfer learning. In IOP Conference Series: Materials Science and Engineering (Vol. 1042, No. 1, p. 012026). IOP Publishing.