# A Performance Efficient Technique for Recognition of Telugu Script Using Template Matching

**N. Shobha Rani**
Maharaja Research Foundation University of Mysore, Maharaja Institute of Technology, Mysore, Karnataka, India
Email: n.shoba1985@gmail.com

**Vasudev T and Pradeep C.H**
Maharaja Research Foundation, University of Mysore, Maharaja Institute of Technology, Mysore, Karnataka, India
Department of Computer Science, Amrita Vishwa Vidyapeetham, Amrita University, Mysore, India
Email: vasu@mitmysore.in, pradeeppanayal@gmail.com

*Abstract*—Feature extraction and classification processes while developing Optical Character Recognition (OCR) systems involve massive computations and quite expensive especially for South Indian scripts. Multiple combinations of vowels and consonants along with its modifiers led to generation of huge number of classes with respect to character recognition systems. The feature extraction and classification of characters from such huge number of classes in south Indian language OCRs remains as a non-trivial problem. This paper proposes a technique for feature extraction and classification of Telugu handwritten script based on customized template matching approach with the support of caching technique for better performance. The technique of caching is implemented using main database with a cache database maintaining the frequently used character templates for 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 on our own test dataset with an overall accuracy of 83.55% for handwritten characters.

*Index Terms*—Telugu characters, Telugu OCR, Customization, template matching, XML schema, cache database.

## I. INTRODUCTION

The offline OCRs imbibe a scanned document input and converts it into machine editable document format necessarily into Unicode of corresponding character images. The input documents are pre-composed with text of either printed or handwritten script pertaining to a particular language. The documents used in character recognition systems are classified as variety of types [1]. The standard classifications are printed and handwritten documents. The printed documents are composed with a particular font style and size of the language. The handwritten documents are written by the author in a particular script, handwriting style of the individual, with the freedom of writing. The freedom and flexibility taken by the author while writing the script adds more complexities in recognition of the characters, since each individual have their own handwriting style and in addition the mistakes while writing appends conflicts in the process of recognition. The combination of printed and handwritten document had resulted into another type of documents called as application/pre-printed documents or forms [2], consisting of both printed and handwritten scripts.

The process of automatic reading of documents is composed of a sequence of stages like image acquisition, pre-processing, object extraction, normalization or windowing, feature extraction, classification and post-processing. The image acquisition is the process of imbibing a document as an input to the character recognition system. Pre-processing stage subjects to removal of noises from the image that occurs due to variety of external factors like improper image scan settings, quality or resolution of image, quality of image capturing device and lack of illumination etc. The resulting images require additional pre-processing techniques like elimination of page layout [3] and graphical components in the image [4], skew detection and correction [5] etc followed by transforming the document to a suitable form for further processing. The object extraction [6] generally called as segmentation; the process of identifying boundaries for region of interest (ROI).

The normalization or windowing is process of converting all the character objects extracted into equal sized blocks [7]. The windowed character objects are further sent for feature extraction [8] in which the features of character object are computed and sent for classification [9]. The classification is the process of determining a match of features computed for particular character extracted with the set of trained character features in the database. The classified character object is considered as the recognized character class which is sent

for post processing where it transforms the character class with the corresponding Unicode of the character. Feature extraction and classification stages being the important concern in the present work, a survey of various techniques of feature extraction is discussed subsequently.

Feature extraction is broadly categorized into low level and high level feature extraction. The low level feature extraction represents the spatial relationships between the pixels which are termed as intensity discontinuities and these techniques do not retrieve the information related to shapes or position or scale etc. in an image. Further low level features represent the edge details in the image which are implemented using first order derivative operators and the second order derivative operators [10]. Some of the popular edge detection operators are Canny, Sobel, Kirsch, Prewitt and Laplacian, Marr-Hildreth operators etc. However, the edge details of particular image are not sufficient enough for recognition of characters in optical systems.

In general high level feature extraction is used for detecting and recognizing the shape of a particular object in the image. Shape description indicates computing the position, orientation and size of a particular object in image. The description of a particular shape depends upon the application requirements. The invariant features of a shape are usually considered to establish the reliability and robustness of the system. The objects in image that seek the shape information requires windowing it to a common size, scale and proper illumination to differentiate the shape from its back ground intensity. Many alternative ways also existing to perform the matching between the test template and trained templates in the images [11-14].

Other sub category with in high level feature extraction techniques includes structural features [15], statistical features [16] and global features [17]. Most shape and texture descriptors fall into this category. These features provide the compact representation of the image and standard classifiers are used to classify the images. However these features are very much sensitive to clutter and noise in the images and require only the presence of a single object. On the other hand characters in south Indian scripts like Telugu are composed of more than one object making it more complex to apply high level feature extraction techniques. Therefore the present work aims at performing the feature extraction and classification of an object using a customized template matching technique. In addition the template matching performance is further improved caching technique and parameter that measures the similarity between trained and test templates.

## II. Related Works

Numerous works are reported in literature using template matching for classification or recognition of characters for different languages. Some of the approaches referred are discussed in brief.

Majid Ziaratban et. al. [21] proposed a language based template matching method for recognition of Farsi/Arabic numerals using neural networks and multi-layer

perceptron model claimed an accuracy of 97.65%. However the experimentation has not considered the complex script for recognition. Sunny Kumar et. al [22] had analyzed the performance of template matching algorithm to English handwritten and type written characters using parameters like accuracy rate and time taken for execution. The accuracy attained is found to be about 83% for the both, however the experiment was carried for a small data set, i.e., on about 360 images, which is relatively less for testing template matching algorithm. Danish et. al. [23] employed the template matching technique for recognition of handwritten, machine printed and type written English characters and numerals obtained an accuracy of about 94.50% for standard typewritten fonts, 88% for unknown type written fonts, 98% for numerals and 75% in case of unknown type written fonts. Nikhil et. al. [24] applied the template for multi font styles and multi font sizes of English script and attained an accuracy of around 90%. Mo Wenying et. al. [25] applied the template matching algorithm by customizing with respect to weighted matching degree. This algorithm provides a higher matching rate and overcome the fallacious recognition produced by traditional calculation method with accuracy of around 100%. Jatin et. al.[26] employed the template matching technique for type written English characters and classified using neural network classifiers. Soumendu et. al. [27] had proposed an algorithm for Japanese character recognition using the centre of gravity features and Euclidean distance features and character with minimum Euclidean distance is the feature employed for charcter recognition. Mahabubar et.al [28] had proposed an method of recognizing Bangla handwritten characters using the convolution neural networks.

In most of the reported works the template matching algorithm is implemented directly for various language scripts and the maximum approaches are focused on English character recognitions. Thus the proposed methodology makes an attempt to analyze the performance of the template matching with required enhancement particular to Telugu character recognition system.

## III. Proposed Methodology

The proposed work aims at providing a customized template matching approach with the improved performance for feature extraction and classification of the Telugu character set. To improve the performance of the template matching algorithm a threshold measure called SIMILARITY MEASURE is imposed along with the shape matching analysis between test and trained templates. Further, to simplify the process of classification an XML database with technique of caching is incorporated. The architecture of the proposed system for feature extraction and classification is represented in fig. 1.
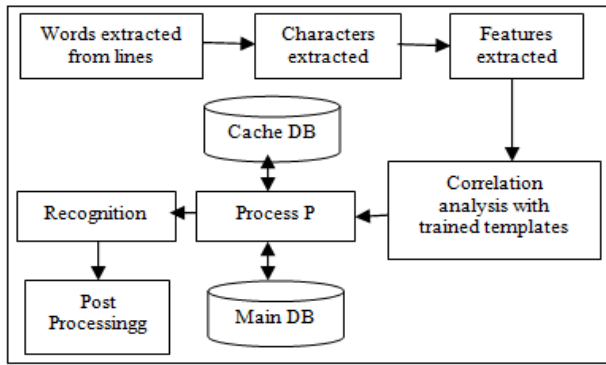
**Process P** →Searches for best correlation between the test and trained template

Fig.1. Architecture of feature extraction and classification

The proposed system proceeds with the input as words extracted from lines and in turn character extraction from each word. The characters extracted are analyzed for its shape by performing the template matching using correlation analysis [25, 26]. The correlation analysis is performed between the trained set of templates and shape extracted from test template for searching the best correlation that satisfies the SIMILARITY MEASURE. Fig. 2(a) and fig. 2(b) depicts samples of the binary templates and edge templates used for shape matching respectively.
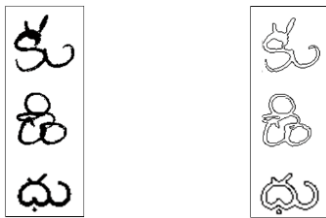


Fig.2(a). Binary templates 2(b) Edge templates

In the usual template matching approaches, the trained template of maximum match among all the templates in the database is considered as the template of best correlation with the test template. The matched template will be further sent for recognition and then for post-processing. The significance of proposed system lies in performing the template matching efficiently. In the perspective of improving the performance of the template matching model, an additional parameter called as SIMILARITY MEASURE is applied to determine the percentage of shape matched between the test template and trained template. Whenever the percentage of correlation between the test and trained template is found to exceed more than a specified threshold then the test template has attained the required amount of SIMILARITY MEASURE and considered as recognized. Thus the implication of SIMILARITY MEASURE reduces the time complexity of traditional template matching process.

Further, the trained templates in the proposed architecture are accessed through a technique of caching by categorizing the entire database into main database and cache database. Caching enhances the performance

of template matching algorithm by classifying the complete set of trained templates between the main database and cache database. The main database is nothing but the XML database of the proposed system consisting of complete set of trained template models. The cache database consists of only references to the recently accessed set of templates in the main database with respect to the particular session of OCR. Whenever a reference is made for analyzing the correlation between the test template and trained template, the initial access is made to cache database if references are available in cache database, then the correlation is computed between the test template and set of recently referenced trained templates available in the cache database. If correlation adhering to SIMILARITY MEASURE threshold is identified from the cache database then it will eliminate the need of performing template matching with the remaining set of templates in the main data base. In case there is no best match identified for test template in the cache database then the correlation analysis is directed towards the trained templates available in the main database. The process of finding a match in main database excludes the set of templates cached in cache database.

*A.  Feature Extraction by Shape Matching*

Feature extraction is the process of representing the images in its compact form so that the features will uniquely identify a particular image. In the proposed work, feature extraction by shape matching is implemented via template matching. Formally the template matching is defined as the method of parameter estimation. The parameters are generally considered as the set of coordinates that represent the positions of the shapes that are extracted from the character templates which is normally the brightness details in the image.

Let F is function of parameter estimation, whose input is the respective coordinate positions of the shape extracted from test template or character. The set of coordinate positions that represent the shape extracted from test template are denoted by

$$S = \left\{ (x_1, y_1), (x_2, y_2), (x_3, y_3)...(x_n, y_n) \right\}$$

and T represents the set of coordinate positions with respect to the shape of trained template and is given by

$$T = \left\{ (p_1, q_1), (p_2, q_2), (p_3, q_3)...(p_n, q_n) \right\}$$

The function F evaluates whether the set S Є T or not. The correlation analysis considers some other parameters like noise in the image. Let the pixel P in the set S is corrupted by Gaussian noise. The noise has a mean of zero and standard deviation is $\sigma$. Thus the probability that a point P, in the set S matching with set T i.e., S(x, y) Є T(p, q) is given by the normal distribution [10] as in equation (3.1).

$$P_{x,y}(p,q) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\left(\frac{T_{p+x,q+y}-S_{p,q}}{\sigma}\right)^2} \tag{3.1}$$

Since the noise affecting each pixel is independent, the probability that the template is at position (p, q) is the combined probability of each pixel that the template covers is given by equation (3.2).

$$L_{p,q} = \prod_{(x,y)\in S} P_{p,q}(x,y) \tag{3.2}$$

By substituting equation (3.1) in (3.2), we have

$$L_{p,q} = \left(\frac{1}{\sqrt{2\pi\sigma}}\right)^n e^{-\frac{1}{2}\sum_{(x,y)\in S}\left(\frac{T_{x+p,y+q}-S_{x,y}}{\sigma}\right)^2} \tag{3.3}$$

Where 'n' is the number of pixels in the template. The function in equation 3.3 is called the likelihood function. Generally, it is expressed in logarithmic form to simplify the analysis. The logarithmic operation scales the function, but it does not change the position of maximum. Thus, by applying logarithm, the likelihood function is redefined as equation (3.4).

$$In(L_{p,q}) = nIn\left(\frac{1}{\sqrt{2\pi\sigma}}\right) - \frac{1}{2}\sum_{(x,y)\in S}\left(\frac{T_{x+p,y+q}-S_{x,y}}{\sigma}\right)^2 \tag{3.4}$$

In maximum likelihood estimation, choose the parameter that maximizes the likelihood function. The trained templates that minimize the rate of change of objective function in equation (3.4) is given by equation (3.5).

$$\min e = \sum_{(x,y)\in S}\left(T_{x+p,y+q} - S_{x,y}\right)^2 \tag{3.5}$$

Hence maximum likelihood estimation is equivalent to choose trained template model in the database that minimizes the squared error. The trained template model for which the test template best matches is the recognized template in the database. Thus the result achieved by template matching is optimal for the test templates that are corrupted by Gaussian noise. The function that maximize the cross correlation between the test template and the trained template through the best match can be computed using equation (3.6).

$$\max e = \sum_{(x,y)\in S}\left(T_{x+p,y+q}S_{x,y}\right) \tag{3.6}$$

The squared error while matching may vary from one trained template to another, the match defined can be poor in case of unmatched templates. Admittedly the

range of correlation is dependent upon the size of the templates and noticeably it is variant to changes in template lighting conditions. In the proposed methodology the template matching is performed with respect to the bit map images of test templates and a set of trained templates. The bitmap image contains the edges of characters inherent in the template. Thus the bitmap image reduces the computation time.

### B. Design of XML Schema for the Creation of Character Classes

XML provides a handy format for the managing the data of any format efficiently and also it is comfortable to integrate with various application programs [19, 20]. This approach provides the advantage of creating a new class structure for representation of huge number of classes and also facilitates the template matching based classification of various characters. We propose a novel XML schema for representation of various classes irrespective of the script processed by OCR. The XML schema includes an index to each class, path specifying the binary edge templates and its corresponding Unicode combination in hexadecimal format. Fig. 3 provides the syntax of a typical XML of class.

```
<Character>
<Index> [Index position in integer format] </Index>
<Letter> [Name of the letter] </Letter>
<Equivalent> [Unicode in UTF-8-Hex format]
</Equivalent>
<Features> [To access the features of binary edge template]
</Features>
```

Fig.3. XML schema for class representation in the database

The character tag <Character> and </Character> is used to enclose the class information like name, path, Unicode required for class creation. The index tag <Index> and </Index> provides the unique index position to each class, the letter tag <Letter> and </Letter> assigns a name to each class which generally represents the image file with .bmp extension of the character template. Equivalent tag <Equivalent> and </Equivalent> encloses the Unicode combination in UTF-8 hexadecimal format and path tag <Features> and </Features> specifies the access path to the features of respective binary edge

```
        <?xml version="1.0"?>
<Characters>
  <Character>
    <Index>1</Index>
    <Letter>- ta</Letter>
    <Equivalent>e0b085</Equivalent>
    <Features>@\telugu\a\- ta.bmp</Features>
  </Character>
  <Character>
    <Index>2</Index>
    <Letter>ya</Letter>
    <Equivalent>e0b0af</Equivalent>
    < Features>@\telugu\a\ya.bmp</Features>
  </Character>
  <Characters>
```

Fig.4. XML database populated with classes

template, which accepts the edge template format in .bmp. Fig. 4 explicates instances of the classes created for the Telugu handwritten and printed OCR in the proposed system.

The <Equivalent> tag in class representation of XML will be able to accept 'n' number of Unicode in hexadecimal format separated by semicolon (;), called as cascading of Unicodes. The feature of cascading multiple Unicode combinations will enable the XML class structure to extent of word level class creation and recognition. The XML class structure is very flexible, generic and suitable to extend for the creation of database of any other script.

### C. Algorithmic Design for Caching

An algorithm has been designed for caching activity during template matching and is presented for comprehension.

$Caching(\tau e_t)$

1. Initialize $TEMP \leftarrow 0$, $\Re eq \leftarrow 0.8$
2. $Identifed\_EQ \leftarrow NULL$
3. Resize $\tau e_t$ to $Dim(55,55)$
4. for n=1 to size(CACHEDB)

$\tau r_t \leftarrow$ CACHEDB{n}. MAINDB

$Act \leftarrow$ CORRELATION($\tau e_t, \tau r_t$)

If( $Act > \Re eq$ )

$TEMP \leftarrow Act$

$Identifed\_EQ \leftarrow$ CACHEDB{n}

PRINT(" TEST TEMPLATE RECOGNIZED IN CACHE DB")

break;

end

end

5. If (TEMP == 0)

for n=1 to size(MAINDB)

$\tau r_t \leftarrow$ MAINDB{n}

$Act \leftarrow$ CORRELATION($\tau r_t, \tau e_t$)

If( $Act > \Re eq$ )

$TEMP \leftarrow Act$

CACHEDB{SIZE (CACHEDB) + 1} $\leftarrow$ MAINDB{n}

$Identifed\_EQ \leftarrow$ MAINDB{n}. EQUIVALENT

MAINDB{n} $\leftarrow$ [ ]

// Marking for restricting repeated access

PRINT("RECOGNIZED IN MAINDB AND ENTRY ADDED TO CACHE DB");

break;

end

end

end

6. If ( $TEMP > \Re eq$ )

//Proceed the test template for post-processing

end

7. Stop.

The $\Re eq$ specifies the percentage of correlation required for assessing the test template to be recognized and $Act$, ACTUAL_SIMILARITY_MEASURE specifies the percentage of correlation of trained template computed. The attribute IDENTIFIED_EQ stores the recognized test template for carrying forward to post processing stage. The trained templates that are recognized in MAINDB will be created as a reference in CACHEDB, further the referenced trained templates will be marked in MAINDB to avoid repeated access in both databases. The scope of the references that are linked in the CACHEDB will be for one particular session of OCR and also will be refreshed again when the MAINDB is reinitialized by the user. The caching and SIMILARITY_MEASURE parameter will improve the performance of the template matching approach effectively, especially when the CACHEDB is very voluminous. Also, employing the SIMILARITY_MEASURE as a parameter to determine the target template also increases the chances of determining occluded or noisy test templates. The figs. 5(a) and 5(b) depicts the obtained similarity measures and recognition results of two images with noise and occlusion respectively.

The $Act$ fig. 5.(a) is estimated as 0.9607243 and the $Act$ of fig. 5.(b) is estimated as 0.8113057.



(a)                    (b)
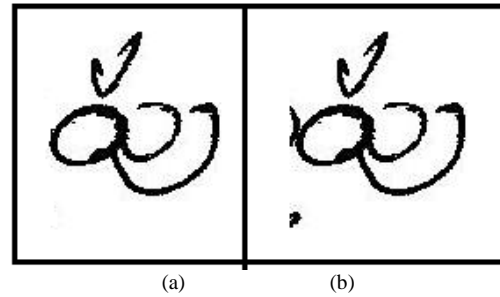
Fig.5. Images with and without occlusion

### IV. EXPERIMENTAL ANALYSIS

Apparently south Indian languages like Telugu consist of a huge alphabetical set include vowels, consonants and consonant modifiers. Telugu language has 16 vowels, 36 consonants and 432 single conjunct composite letters (Guninthalu) of 504 after excluding the vowels with ardaanuswara (ఁ) and anuswara (ం) along with the mere consonant (క్). The handwritten datasets for the experimentation had been collected from 25 authors each comprising of vowels, consonants, single conjunct and double conjunct vowel consonant clusters.

The training to the database is conducted with 2730 handwritten character templates. The datasets used for training includes 2730 frequently used character set of Telugu script of 6 different users and test set of size 455 characters from one author. The 455 character templates include vowels, consonants and single as well as multi-conjunct vowel consonant clusters. The frequently used character set is composed with reference to the data used in filling Telugu application form documents. The documents considered belong to the geographical

locations of Anantapur district, Andhra Pradesh. The selection of frequently used character set for OCR results in an optimal speed and accuracy.

Some of the frequently used single conjunct and double conjunct vowel-consonant clusters are considered as individual classes in the present work. The creation of separate classes for double conjunct vowel-consonant clusters eliminate the need of maintaining symbolic ordering during segmentation and post-processing stages, this avoids the number of iterations required for classification of each conjunct, vowels and consonants separately. And also post-processing of double conjunct vowel-consonant clusters may lead to erroneous recognition results if proper symbolic ordering is not followed while classification. However the increase in number of classes may increase the search time involved in recognition of characters. But in the proposed system a caching technique is incorporated to suffice the performance of the system. The fig. 6 represents the multi conjunct vowel-consonant clusters and the fig. 7 represents some of the double conjunct consonant clusters of handwritten datasets.



Fig.6. Multi conjunct vowel consonant clusters



Fig.7. Double conjunct vowel-consonant clusters

An user friendly interface has been designed using MATLAB for performing the character recognition. The GUI includes the functionalities to load the input documents, to refresh the cache database which updates the total number of classes present at current instance of execution and the new button to reload another document input.

The results of experimentation are tabulated in table 1 and the REQUIRED_SIMILARITY_MEASURE $\Re eq$ considered for the best correlation is 80%.

The accuracy obtained with test templates is around 93.55% for the sample size of around 2730 different characters from which 2554 characters are recognized correctly.

The fig. 8 graphically represents the performance of caching technique for the data statistics represented in Table 2.

Table 1. Experimental results of trained and test templates

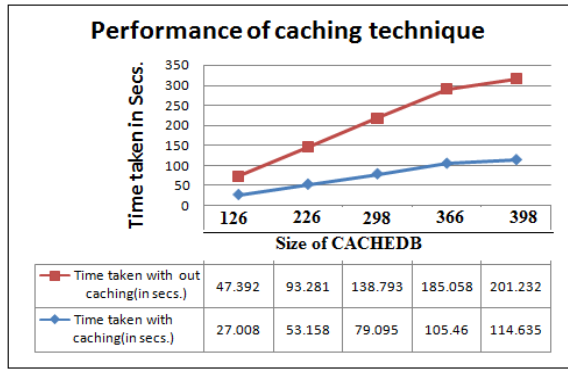| Bitmap image of character | Required similarity measure | Recognized Character | Obtained similarity measure | Recognized Charact |
|---|---|---|---|---|
| ఒ | 9.647845e-001 | ఒ | 8.343535e-001 | ఒ |
| ఒ | 9.598691e-001 | ఒ | 8.297401e-001 | ఒ |
| ఋ | 9.566974e-001 | ఋ | 8.479653e-001 | ఋ |
| ఋ | 9.467432e-001 | ఋ | 8.234562e-001 | ఋ |
| ఉ | 9.746453e-001 | ఉ | 8.325270e-001 | ఉ |
| ఊ | 9.736098e-001 | ఊ | 8.620125e-001 | ఊ |
| ఎంు | 9.713786e-001 | ఎంు | 7.523467e-001 | ఊ |
| ఎ | 9.609760e-001 | ఎ | 8.873520e-001 | ఎ |
| ఏ | 9.553150e-001 | ఏ | 8.642541e-001 | ఏ |
| ఒ | 9.584122e-001 | ఒ | 8.693013e-001 | ఒ |
| ఓ | 9.592538e-001 | ఓ | 8.481125e-001 | ఓ |
| క్తూ | 9.288411e-001 | క్తూ | 8.440577e-001 | క్తూ |
| క్తి | 9.367023e-001 | క్తి | 8.338047e-001 | క్తి |
| క్తూ | 9.325272e-001 | క్తూ | 8.443321e-001 | క్తూ |
| కె | 9.464063e-001 | కె | 8.456852e-001 | కె |
| కే | 9.352383e-001 | కే | 8.420255e-001 | కే |
| కై | 9.472471e-001 | కై | 8.489905e-001 | కై |
| కొ | 9.415977e-001 | కొ | 8.484739e-001 | కొ |
| కో | 9.472471e-001 | కో | 8.367062e-001 | కో |
| కౌ | 9.578262e-001 | కౌ | 8.549644e-001 | కౌ |

Fig.8. Size of CACHEDB versus time taken with/without caching

The size of CACHEDB represents the number of templates cached in cache database.

Table 2. Experimental statistics considered for evaluation of caching technique

| No. of documents | Size of MAINDB | Size of CACHEDB | No. of handwritten characters recognized |
|---|---|---|---|
| 1 | 2730 | 126 | 126 |
| 2 | 2730 | 226 | 248 |
| 3 | 2730 | 298 | 369 |
| 4 | 2730 | 366 | 492 |
| 5 | 2730 | 398 | 535 |

From the fig. 8 and table 2, it is evident that the increase in number of documents is directly proportional to increase in size of CACHEDB. The increase in size of cache represents the new character templates that are referenced from MAINDB and linked to cache, whenever a new document is considered for processing during a session.

It is observed that recognition accuracy remains same with or without caching technique in the proposed system and only characteristic that varies is performance of the system. It is recommended to process similar set of documents in one particular session of OCR to attain optimal performance of the system. The user needs to refresh the CACHEDB whenever the subsequent document to be processed are found to be completely different from the one so far processed. Refreshing the CACHEDB flushes all the entries in CACHEDB or sets the size of CACHEDB to zero.

In the proposed system the experimentation is also extended to one set of references of printed characters and its performance is indicated in fig. 9.
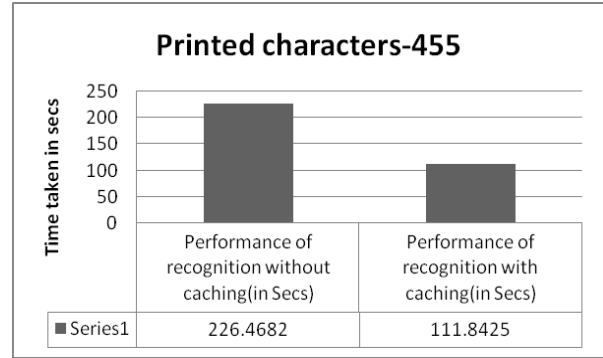


Fig.9. Performance of recognition – Printed Characters

The testing of caching technique is performed on windows platform with core i5 processor and 4 GB RAM. The higher configuration in machine may definitely result a better performance. The performance of recognition through caching can be improved by testing with more number of similar documents. This is due to the fact that the size of cache database will be increased in par with the increase in number of input documents used for recognition. In the same way number of documents used for recognition in a particular session automatically increases the frequently used templates in cache database. In the cases of documents consisting of non-frequently or non-repeatedly occurring textual patterns the performance of recognition depends on the size of MAINDB.

## V. CONCLUSIONS

The feature extraction by shape matching in conjunction with correlation based classification has provided satisfactory results. The inclusion of REQUIRED_SIMILARITY_MEASURE to find the suitable match between the test template and trained template greatly reduces the worst case time complexity of conventional template matching technique. However there are few cases of misrecognition especially for some of the confusing character pairs [కో, కో], [ఎ, ఏ] [ట, ఒ] [ఒ, ఓ] etc. The confusing characters possess very minor difference in its structural orientation, but the use of REQUIRED_SIMILARITY_MEASURE improves the performance of template matching algorithm. However the design of 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 experimental results of the proposed system can be still improved by replacing template matching with an efficient feature extraction technique so as to reach high recognition accuracies.

REFERENCES

[1]  Rinki Singh, Manideep Kaur, OCR for Telugu Script Using Back-Propagation Based Classifier, International Journal of Information Technology and Knowledge Management, Vol. 2, No. 2, 2010, pp. 639-643.

[2]  Suman V Patgar, Vasudev T, Murali S, A system for detection of fabrication in photocopy document, Journal of Computer Science & Information Technology, Vol. 5, No. 14, 2015, pp. 29–35.

[3]  B. Verma, M. Blumenstein, S. Kulkarni, Recent achievements in off-line handwriting recognition systems, School of Information Technology, Griffith University, Gold Coast Campus.

[4]  C. V. Jawahar, M. N. S. S. K. Pavan Kumar, S. S. Ravi Kiran, A Bilingual OCR for Hindi-Telugu Documents and its Applications, Centre for Visual Information Technology, International Institute of Information Technology, Hyderabad.

[5]  N. Shobha Rani, T. Vasudev, "A Generic Line Elimination Methodology using Circular Masks for Printed and Handwritten Document Images ", Emerging research in computing, information, communication and applications ELSEVIER science and technology, Vol. 3(1), 2014, pp. 589-594.

[6]  Rangachar kasturi, Lawrence O'gorman, Venu govindaraju, Document image analysis: A primer, Sadhana, Vol. 27, No. 1, 2002, pp. 3–22.

[7]  Nallapareddy Priyanka, Srikanta Pal, Ranju Mandal, Line and word segmentation approach for printed documents, IJCA Special Issue on Recent Trends in Image Processing and Pattern Recognition, Vol. 1 2010, pp. 30-36.

[8]  R. Zhang, T. T. Tjhung, Senior Member, IEEE, H. J. Hu, and P. He, Window Function and Interpolation Algorithm for OFDM Frequency-Offset Correction, IEEE transactions on vehicular technology, Vol. 52, No. 3, 2003, pp. 654-670.

[9]  Ryszard S. Choras, Image Feature extraction techniques and their applications for CBIR and Biometrics Systems, International journal of biology and biomedical engineering, Vol. 1, No. 1, 2007, pp. 6-16.

[10]  Pooja Kamavisdar, Sonam Saluja, Sonu Agrawal, A Survey on Image Classification Approaches and Techniques, International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, No. 1, 2013, pp. 2278-1021.

[11]  Mark Nixon, Alberto Aguado, Feature extraction and Image processing", Second Edition, Academic Press, Elsevier, London, UK, 2002.

[12]  Yafang Xue, Optical Character Recognition, Department of Biomedical Engineering, University of Michigan, 2014.

[13]  Avneet Kaur, Lakhwinder Kaur, Savita Gupta, Image Recognition using Coefficient of Correlation and Structural SIMilarity Index in Uncontrolled Environment, International Journal of Computer), Vol. 59, No.5, 2012, Applications pp. 0975 – 8887.

[14]  Jangala. Sasi Kiran, N. Vijaya Kumar, N. Sashi Prabha, M. Kavya, A Literature Survey on Digital Image Processing techniques in character recognition of Indian languages, International Journal of Computer Science and Information Technologies, Vol. 6, No. 3, 2015, pp. 2065-2069.

[15]  M. Shalini, B. Indira, Automatic Character Recognition of Indian Languages – A brief Survey, International Journal of Innovative Science, Engineering & Technology, Vol. 1 No. 2, 2014, pp. 131-138.

[16]  E.Kavallieratou, N.Fakotakis and G.Kokkinakis, Handwritten Character Recognition based on Structural Characteristics, Proceedings of the 16th International Conference of the International Association of Pattern Recognition, Vol. 3, 2002, pp. 139-142.

[17]  S. Arora, D. Bhattacharjee, M. Nasipuri, D. K. Basu, M. Kundu, Application of statistical features in handwritten Devanagari character recognition, International Journal of Recent Trends in Engineering, vol. 2, 2009, pp. 40–42.

[18]  Dimitri A. Lisin, Marwan A. Mattar, Matthew B. Blaschko, Mark C. Benfield, Erik G. Learned-Miller, Combining Local and Global Image Features for Object Class Recognition, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 3, 2005, pp. 47-55.

[19]  John D. Evans, XML for Image and Map Annotations (XIMA) Draft Candidate Interface Specification, Version 0.4, NASA Digital Earth Office, Goddard Space Flight Center, Greenbelt, 2001.

[20]  Swapnil Behle, Development of OHWR system for Hindi, C-DAC, Pune, 2013.

[21]  Majid Ziaratban, Karim Faez, Farhad Faradji, Language based feature exraction using template matching in Farsi/Arabic handwritten numeral recognition, Ninth International Conference on Document Analysis and Recognition, vol. 1, 2007, pp. 297–301.

[22]  Sunny Kumar, Pratibha Sharma, Offline Handwritten & Typewritten Character Recognition using Template Matching, International Journal of Computer Science & Engineering Technology, Vol. 4 No. 06, 2013, pp. 818-825.

[23]  Danish Nadeem, Saleha Rizvi, "Character recognition using template matching", Project report, Department of Computer Science, Jamia Millia Islamia, New Delhi, 2015.

[24]  Nikhil Rajiv Pai, Vijaykumar S. Kolkure, Design and implementation of optical character recognition using template matching for multi fonts /size, International Journal of Research in Engineering and Technology, Vol. 4, No. 2, 2015, pp. 398-400.

[25]  Mo Wenying, Ding Zuchun, A Digital Character Recognition Algorithm Based on the Template Weighted Match Degree, International Journal of Smart Home, Vol. 7, No. 3, 2013, pp. 53-60.

[26]  Jatin M Patil, Ashok P. Mane, Multi Font And Size Optical Character Recognition Using Template Matching, International Journal of Emerging Technology and Advanced Engineering, Vol. 3, No. 1, 2013, pp. 504-506.

[27]  Soumendu Das, Sreeparna Banerjee, An Algorithm for Japanese Character Recognition, International Journal of Image, Graphics and Signal Processing (IJIGSP), Vol. 7, No. 1, 2014, pp. 9-15.

[28]  Md. Mahbubar Rahman, M. A. H. Akhand, Shahidul Islam, Pintu Chandra Shill, M. M. Hafizur Rahman, Bangla Handwritten Character Recognition using Convolutional Neural Network, International Journal of Image, Graphics and Signal Processing (IJIGSP), Vol. 7, No. 8, 2015, pp. 42-49.

## Authors' Profiles

**N. Shobha Rani** is currently working as Assistant professor in Amrita Vishwa Vidyapeetham University, Mysore an also pursuing her Ph.D degree in Maharaja Research Foundation, University of Mysore, mysore. Her areas of interest include Document image processing, Optical character recognition and Computer vision.

**Dr. Vasudev T** is Professor, in the Department of Computer Applications, Maharaja Institute of Technology, Mysore. He obtained his Bachelor of Science and post graduate diploma in computer programming with two Masters Degrees one in Computer Applications and other one is Computer Science and Technology. He was awarded Ph.D. in Computer Science from University of Mysore. He is having 30 years of experience in academics and his area of research is Digital Image Processing specifically document image processing and authored about 50 research papers

**Pradeep C.H** is pursuing his MCA degree in department of Computer Science, Amrita Vishwa Vidyapeetham, Mysore. He is placed in Payoda technologies, Coimbatore as a software engineer. His areas of research include Digital image processing and pattern recognition.