ISSN: 0974-5572

IJCTA

International Journal of
CONTROL THEORY
AND APPLICATIONS

UGC APPROVED JOURNAL

ज्ञान-विज्ञान विमुक्तये

INTERNATIONAL SCIENCE PRESS
Gurgaon, Haryana (India)

International Journal of Control Theory and Applications

ISSN : 0974-5572

© International Science Press

Volume 9 • Number 46 • 2016

Recognition Method of Text CAPTCHA using Correlation and Principle Component Analysis

Navjot Rathour[a] and Vinay Bhatia[b]

[a]*Department of Electronics and Communication Engineering, Lovely Professional University, Phagwara, 144402, India. Email: er.rathour@gmail.com*
[b]*Baddi University of Emerging Sciences and Technology, Baddi, 173205, India*

Abstract: A Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) is a type of challenge-response test used in computing as an attempt to ensure that the response is generated by a person. These are typically implemented as distorted text which the user must correctly transcribe. Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) should be easy for a machine to automatically generate, easy for a human to solve, and difficult, or impossible, for a machine to solve even if the generation algorithm is publicly available. Online service providers like PayPal use Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) to prevent automated signups and abuse of their services. This paper presents a Principal Component Analysis (PCA) and cross correlation based approach in breaking of PayPal Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA). The process can be broken down into 3 tightly coupled tasks; namely pre-processing, segmentation and recognition. The pre-processing of the image is performed to remove all the background noise of the image. A new segmentation algorithm has been presented that will first detect the connectivity; remove the connectivity and gives the segmented output. The recognition is performed by using the correlation values of the inputs and templates. The second method used for recognition is a Principal Component Analysis (PCA) the success rate of recognition using correlation is 90% and the success rate using Principal Component Analysis (PCA) is 97%.

Keywords: Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA), Optical character recognition (OCR), Principal Component Analysis (PCA), Image Processing.

1. INTRODUCTION

A Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) is a program that can generate and grade tests that humans can pass but current computer programs cannot. For example, humans can read distorted text as the one shown below, but current computer programs are not able to do so. The term Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) was coined in 2000 by Luis von Ahn, Manuel Blum, Nicholas Hopper and John Langford of Carnegie Mellon University.

At the time, they developed the first Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) to be used by Yahoo recognition[1] In this paper a new algorithm for segmentation of text Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) particularly used by the online service provider PayPal has been designed and performed for the detection and separation of the straight and diagonal connectivity. Three different Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) including the PayPal Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) are shown in Figure 1.

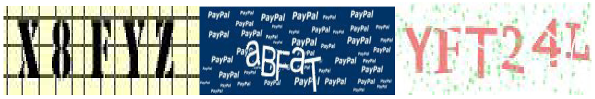


Figure 1: Three CAPTCHAs based on text (left to right): PayPal, PayPal, LPU

1.1. Optical Character Recognition

Optical character recognition, usually abbreviated to OCR, is the mechanical or electronic translation of scanned images of handwritten, typewritten or printed text into machine encoded text[2]. It is widely used to convert books and documents into electronic files, to computerize a record keeping system in an office, or to publish the text on a website. The Optical character recognition (OCR) process can be broken down into three steps: preprocessing, segmentation and classification[9].

1.2. Preprocessing

Preprocessing is the initial stage of Optical character recognition (OCR). The preprocessing step can be omitted if the image is noise free. However Optical character recognition (OCR) is mainly used for scanned documents that generally contain noise from bad paper, printer or scanner. So preprocessing is must in order to remove that noise. Preprocessing generally consist of noise removal, skew correction and threshold. Noise can be removed by filtering the image for removing the stray or unwanted marks. Skew correction can be performed by estimating the angle of line of text[9]. Threshold is often used to convert an image into binary image. Threshold works well in the case where noise and information varies widely in terms of intensity.

1.3. Segmentation

Segmentation is the method of decomposing the complete image into individual text images. Segmentation can be performed based on the local decision made on the basis of shape or similarity, as well as global decision with regards to surrounding context. Segmentation is the most critical and most important step in Optical character recognition (OCR) because the maximum possibility of error in Optical character recognition (OCR) is at segmentation part only[3]. In 1996, Richard Casey and Eric Lecolinet surveyed the available methods and defined three categories for offline character segmentation methods, based on how segmentation and classification interact in the overall process[5]:

1. Dissection Approach: a single partitioning of the image into sub images based on character-like" properties followed by classification of the sub images[5].
2. Recognition-based Approach: segmentation where the image is iteratively searched for components that most closely match the classes in the alphabet [5].
3. Holistic Approach: segment and recognize words as single units (character segmentation is avoided)[5].

The segmentation of characters varies in difficulty based on the input type. It is easy to segment the characters with a fixed pitch i.e. generally in machine printed text of pixel pitch. The difficulty of segmentation increases where the pitch is not fixed like in cursive handwriting or in real time images of text. So for such segmentation more complex approach is required. Advanced techniques like Hidden Markov Models (HMMs) and Artificial Neural Networks (ANNs) are used In order to achieve high accuracy on complex problem domains, segmentation and recognition cannot be treated independently [6].

Classification

Character classification is typically performed based on feature vectors. Feature extraction is the process of transforming the input data into a reduced representation. It is commonly employed when there is too much input data to efficiently process or if the input data is redundant (lots of data but not much information). This simplification of the input data provides an accurate description of a larger set of data. The set of application-dependent features are typically chosen by domain experts. However if no such experts exist, other dimensionality reductions, such as principle component analysis (PCA), can still be performed [8].

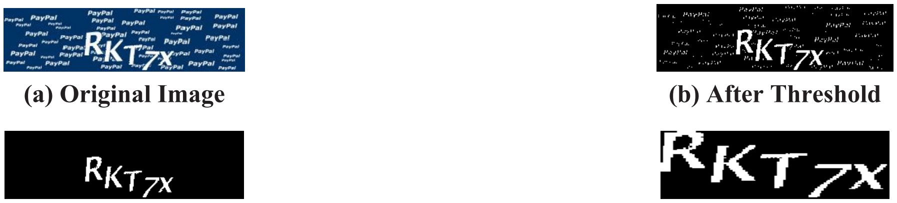
Principle Component Analysis (PCA) is used to reduce the dimensionality of multi-dimensional data sets. It works by removing characteristics about the data which have low impact on the variance of the overall dataset. Similarly, it attempts to retain characteristics which contribute most to its variance. Once the feature vectors are computed, classification can be performed. Classification can be done by nearest neighbour techniques based on templates, neural networks[6], etc. The classification mechanism is largely non-important in the resulting decision. The most important step is the feature extraction, of which many methods exist[4]. Naive methods feed entire image matrices, while other requires experts to develop visual cues to distinguish characters from one another. Depending on the problem domain, classifiers can be aided by the inclusion of a dictionary or N-gram statistics.

2. METHODOLOGY

The breaking of PayPal Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) is done on the Optical Character Recognition (OCR) based approach i.e. Preprocessing, segmentation and recognition based three step approach is used.

2.1. Preprocessing

Preprocessing is essentially used to remove the noise. Generally noise is added in the Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) in order to make them difficult to understand by Optical Character Recognition (OCR) system. So this noise must be removed, only then the success in segmentation and classification level can be achieved. The first step in our processing is to apply threshold to remove backgrounds unwanted lines or noise. For that purpose the image is first converted into grayscale image and after that image is Threshold. Threshold value remains same for all the Completely Automated Public Turing



(a) Original Image

(b) After Threshold

(c) After cleaning

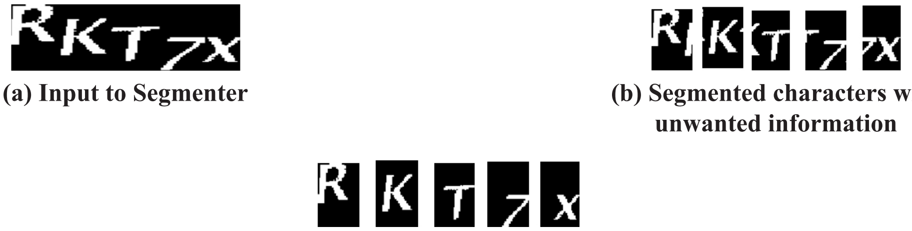
(d) After Bounding

Figure 2: Preprocessing Step

Test to Tell Computers and Humans Apart (CAPTCHA) as the PayPal uses same type of Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) with random text only. After threshold the image is detected for any other noise in terms of unwanted on pixels and removed in the step of cleaning. After cleaning we have noise free binary image which is bounded in a box using the region properties.

2.2. Segmentation

Segmentation is the second step of our process. Once the noise from the image is removed the image is segmented into sub images. As we know that segmentation is one of the most important and critical step in the process of breaking Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA). One of the simplest approach that can be used for segmentation is to crop the individual character based on the area occupied by each character, the result of this approach are shown in following Figure 3. In this we have separated the characters and by cropping each character more than its area. The extra part of another character that comes with the desired character is removed by its area and finally the segmented individual character obtained[8].



(a) Input to Segmenter

(b) Segmented characters with unwanted information

(c) Final segmented characters

Figure 3: Segmentation Step

As in Figure 3(c) we have seen the segmented characters for a fixed sized and unconnected image but it is difficult to make this type of segmentation dynamic because the size of the Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) is not same for all images and different types of connectivity has also been identified between the Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) characters. So the results with this algorithm are not correct for all the Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA). Figure 4 shows the results that are not appropriate with this algorithm.

(a) Input to segmenter

b) Segmented characters

Figure 4: Wrong segmentation results

2.2.1. New Algorithm for Segmentation

A new algorithm has been designed for the dynamic segmentation of all the characters present in Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA). It was not always the case that we always found connectivity between the characters in Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA). So the segmentation of such Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) is done on the basis of area occupied by each character. In this algorithm we first check the area of individual character and removing it one by one by keep on subtracting it from previous image as shown in Figure 5 the process is repeated till all the characters get separated.

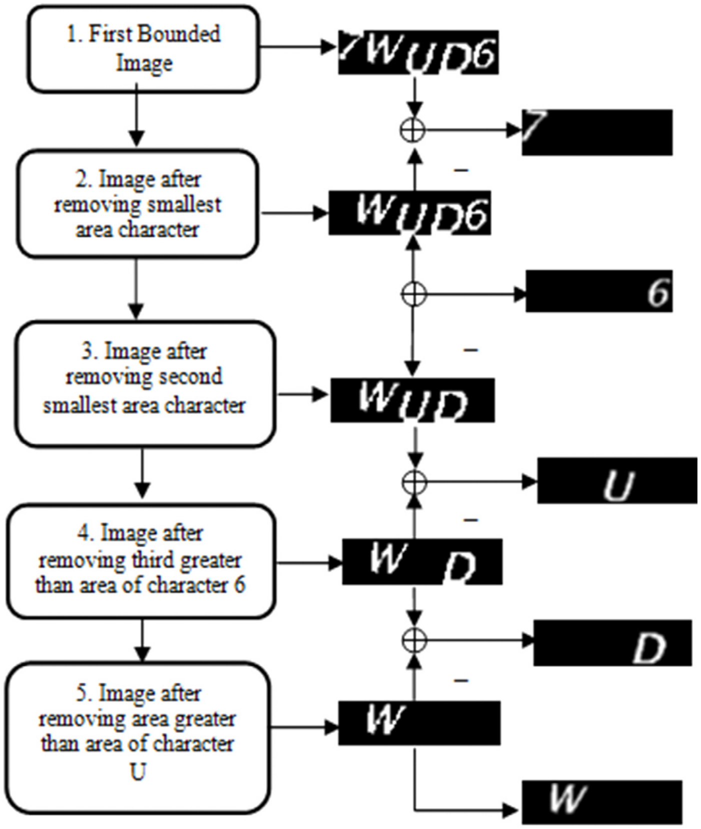

1. First Bounded Image
2. Image after removing smallest area character
3. Image after removing second smallest area character
4. Image after removing third greater than area of character 6
5. Image after removing area greater than area of character U
7WUD6
7
WUD6
6
WUD
U
W D
D
W
W
−
−
−
−

Figure 5: New segmentation algorithm

The segmentation algorithm starts with the bounded image and can be explained with the following set of equations

$$I_{A(i,j)} = I_{Bounded(i,j)} \tag{1}$$

$$I_{B(i,j)} = I_{A(i,j)} - (I_{A(i,j)\text{ smallest area character}}) \tag{2}$$

$$I_{C(i,j)} = I_{B(i,j)} - (I_{B(i,j)\text{ smallest area character}}) \tag{3}$$

$$I_{D(i,j)} = I_{C(i,j)} - (I_{C(i,j)\text{ smallest area character}}) \tag{4}$$

$$I_{E(i,j)} = I_{D(i,j)} - (I_{D(i,j)\text{ smallest area character}}) \tag{5}$$

Above equations gives the final segmented images with individual character in single images by applying the following set of equations on the above equations.

$$I_{1(i,j)} = I_{A(i,j)} - I_{B(i,j)} \tag{6}$$

$$I_{2(i,j)} = I_{B(i,j)} - I_{C(i,j)} \tag{7}$$

$$I_{3(i,j)} = I_{C(i,j)} - I_{D(i,j)} \tag{8}$$

$$I_{4(i,j)} = I_{D(i,j)} - I_{E(i,j)} \tag{9}$$

$$I_{5(i,j)} = I_{E(i,j)} \tag{10}$$

In the above equations $I_{1(i,j)}$, $I_{2(i,j)}$, $I_{3(i,j)}$, $I_{4(i,j)}$ and $I_{5(i,j)}$ are the final segmented images. Now the next step is to put these images in bounded box in order to remove the unwanted area.

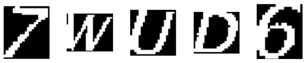


Figure 6: Separated and ordered characters in ascending order

The next step is to arrange all these separated images in proper order. This can be done by calculating the centeroid of the individual separated images and arrange them in ascending order. The results after arranging the images in ascending order are shown in Figure 6. As in Figure 6,the size of the each individual character is different so to make the recognition of each character we have resized all the separated characters of the Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA). The result after resizing is shown in Figure 7.

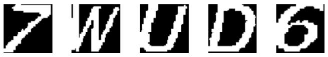
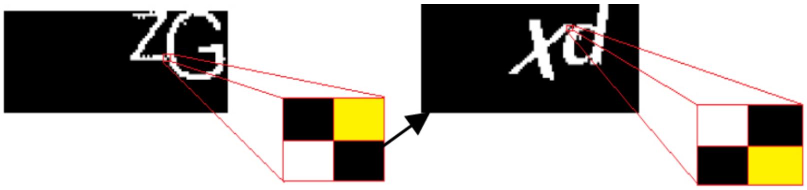


Figure 7: Resized and ordered characters

2.2.2. Identification of Connectivity

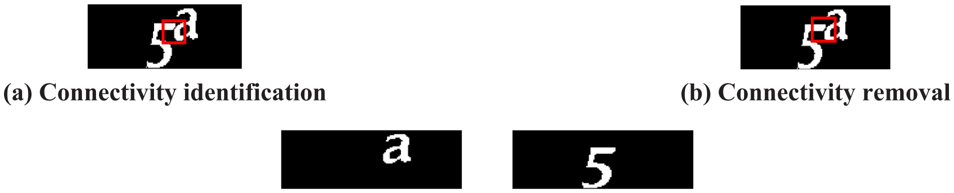
In case of the Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) where the characters have connectivity with each other, we first have to identify the type of connectivity and separate those pixels to get individual characters. The majority of the Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) are found to have diagonal connectivity. Two types of diagonal connectivity has been identified as shown in the figure 8. An algorithm has been designed to detect the both type of diagonal connectivity and separate the characters[10].

The Boxes which are yellow are indicating the pixel that will be removed after identification of connectivity. Using this algorithm we have first identified the connectivity and then removed the connectivity. Figure 9 shows the results of this algorithm.



(a) Upper diagonal connectivity **(b) Lower diagonal connectivity**

Figure 8: Diagonal connectivity identification and separation algorithm



(a) Connectivity identification **(b) Connectivity removal**

(c) Separation of characters after removal of connectivity in characters

Figure 9: Results of connectivity identification and separation algorithm

Mathematically the above algorithm can be understood by the following set of windows which are showing upper and lower diagonal connectivity.

The other type of connectivity that has been identified in few Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) is straight connectivity. Figure 10 shows the straight connectivity

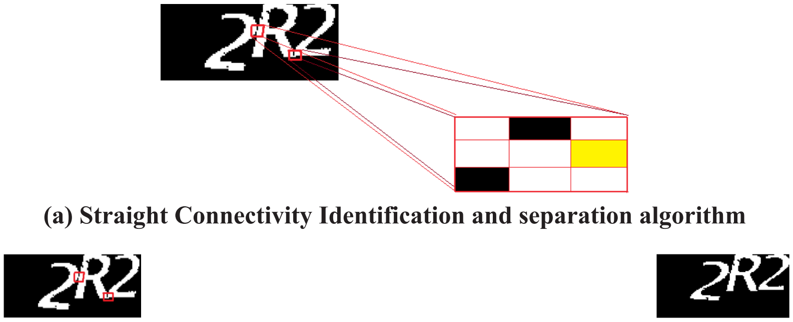
$$I_{DUpper(i,j)} = \begin{array}{|c|c|}\hline 0 & 1 \\ \hline 1 & 0 \\ \hline\end{array}$$

(a) Upper diagonal connectivity window

$$I_{DLower(i,j)} = \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & 1 \\ \hline\end{array}$$

(b) Lower diagonal connectivity window

Figure 10: Windows for diagonal connectivity



(a) Straight Connectivity Identification and separation algorithm

(b) Connectivity identification

(c) Connectivity removal

Figure 11: Results of straight connectivity identification and removal

1	0	1
1	1	1
0	1	1

Figure 12: Window for straight connectivity

in the Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) image. An algorithm has been designed to detect the straight connectivity and the separation of such type of connectivity is done by simply removing the pixel shown yellow in color[10].

2.3. Recognition

The recognition of the characters after segmentation is done by two different methods for comparison purpose. We performed the recognition with both PCA and templates correlation. For template correlation we made a template matrix of dimension (40 × 40 × 62). The Template Matrix consist of 62 values i.e. from 0 to 9, *a* to *z* and A to Z but the values like '0', '1', 'o', 'i', 'l', 'I', 'O', 'Q' are empty because these characters are not used by the PayPal Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA). The methods used for making template matrix consist of the same steps that are used for pre-processing and segmentation of the Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) images [8].The recognition of the PayPal Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA) by using correlation gives confidence of 90%.Figure 11 gives the correlation confidence for "gFFNT". The template correlation classifier calculates the 2 Dimensional (2D) correlation coefficients for the input image I and the template T. The index of the template with the highest 2 Dimensional (2D) correlation coefficient with the input image is returned as the match. The other method used for recognition is Principal Component Analysis (PCA). The steps were used for recognition using Principal Component Analysis (PCA) are:

Step 1: The very first step for Principal Component Analysis (PCA) is to make the centeroid of the entire data equal to zero. This can be done by subtracting the mean.

Step 2: The Second step consists of calculating the Covariance Matrix.

Step 3: The next step after finding covariance matrix is to find the Eigen values of covariance matrix.

Step 4: The next is for choosing the components and forming the feature vector. In order to form feature vector we need to order the Eigen vectors according to Eigen value with highest to lowest priority.

Step 5: This is how we derive a new set of data which gives us the original data solely in terms of vectors we chose. Eigen vectors are now the axes.

2.3.1. Database for Principal Component Analysis (PCA)

A square, N by N image can be expressed as an N^2 dimensional vector as represented by (11).Were the rows of pixels in the image are placed one after the other to form a one dimensional image. E.g. the first N. elements (x_1, ..., a_n) will be the first row of the image, the next N elements are the next row, and so on. The values in the vector are the intensity values of the image, possibly a single greyscale value. Say we have 20 images.

$$X = (X_1 , X_2, X_3, \ldots, X_N^2) \tag{11}$$

Each image is N pixels high by N pixels wide. For each image we can create an image vector as described in the representation section. We can then put all the images together in one big image-matrix.

$$\text{Image Matrix} = \begin{bmatrix} \text{Image Vec 1} \\ \text{Image Vec 2} \\ \text{Image Vec 3} \end{bmatrix} \tag{12}$$

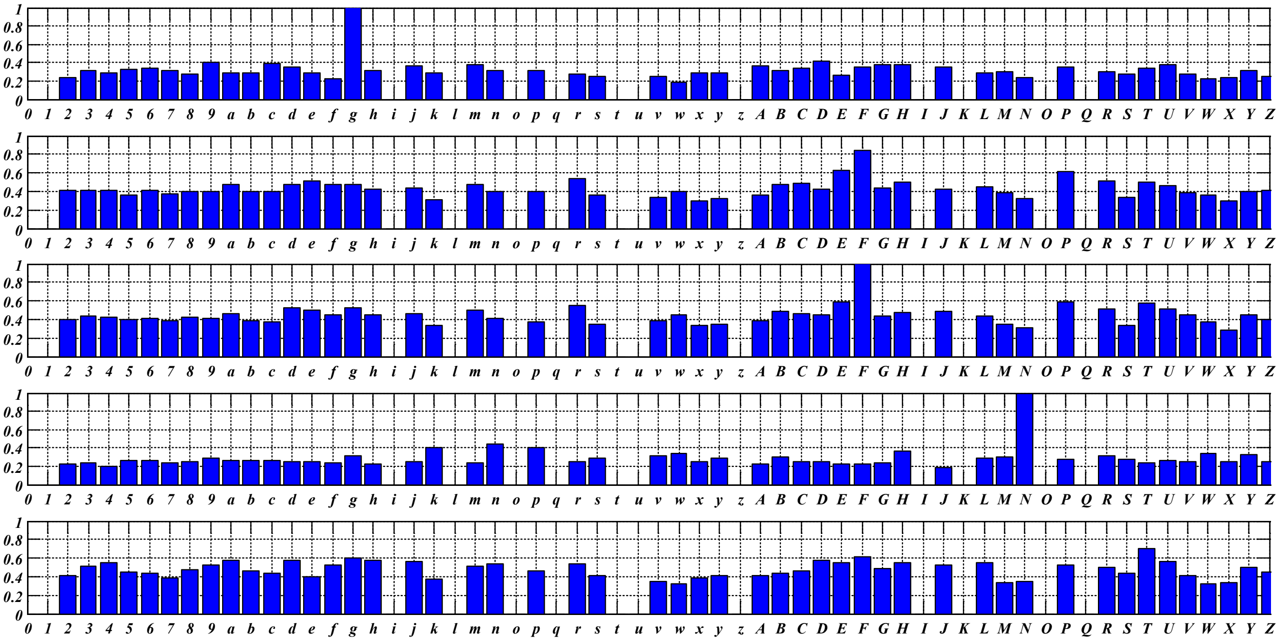
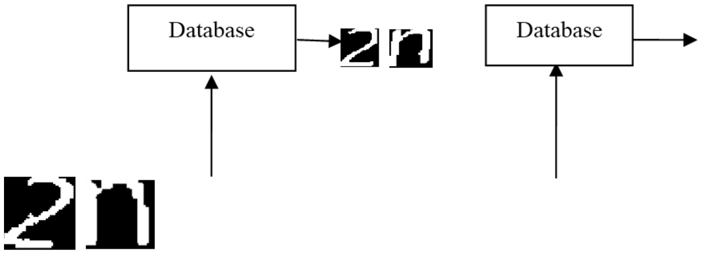


Figure 13: Template Correlation confidences for "gFFNT"

As given in equation (12) a starting point for our Principal Component Analysis (PCA). Once we have performed Principal Component Analysis (PCA), we have our original data in terms of the eigenvectors we found from the covariance matrix. In future research, all types of connectivity shall be targeted for improvement to increase the recognition rates. Firstly algorithms to identify all types of connectivity need to be designed efficiently. The algorithm was designed to detect the straight and diagonal connectivity. The main problem lies with the segmentation that can be efficiently carried out by separating the connected components.



(a) Recognition of '2' using PCA **(b) Recognition of 'n' using PCA**

Figure 14: Results of Recognition using PCA

3. CONCLUSION

Through this paper the authors have presented an efficient algorithm to segment the characters and to recognize them using correlation method as well as with Principal Component Analysis (PCA). The templates were used for cross correlation and results were 90% with these templates. The recognition results with Principal Component Analysis (PCA) come out to be equal to 97%.The type of connectivity that has been identified and separated here is only for straight and diagonal. Some research work also indicate segmentation results after removing connectivity also which would form basis of further improvements in this direction.

REFERENCES

[1] F Shin-Yu Huang, et. al. A projection based segmentation algorithm for breaking MSN and Yahoo CAPTCHAS. Proceedings of World Congress on Engineering. Vol. 1, July 2008.

[2] Wikipedia Optical character recognition-wikipedia, the free encyclopedia, 2012.

[3] I. S. Jacobs and C. P. Bean Richard, L. Hoffman and J. warren Mc-Cullough. Segmentation methods for recognition of machine printed characters. IBM Journal of Research and Development. 1971 March,p.153-165.

[4] Qivind Due Trier, Anil K. Jain and Torfinn Taxt, Feature extraction methods for character recognition-a survey, Pattern Recognition;1996.p. 641–662.

[5] Richard G. Casey and Eric Lecolinet, A survey of methods and strategies in character segmentation, IEEE Trans. Pattern Analysis and Machine Intelligence;1996 July. p. 690–706.

[6] Mohammad Osiur Rahman et. al, Real time road sign recognition system using artificial neural networks for bengali textual information box, EJSR; 2009. p.478-487.

[7] Bin Yu and Anil K. Jain. A generic system for form dropout. Transactions on Pattern Analysis and Machine Intelligence; 1996;18(11):1127-113.

[8] Kurt Alfred Kluever, Independent study report character segmentation and classification, Department of Computer Science, Golisano College of Computer and Information Sciences, Rochester Institute of Technology, February 28, 2008.

[9] L.von Ahn, M. Blum, and J. Langford. Telling humans and computers apart (automatically). CMU Tech Report CMU-CS-02-117, February 2002.

[10] Alman Amin Khan, Character segmentation heuristics for check amount verification [Masters thesis]. Massachusetts Institute of Technology; 1998, June.