Audio Enabled Image Text Translation and Reading

[1] Aditi N. Kulkarni, [2] Prof S. S. Nagtilak

[1]Student, [2]Professor
[1] Dept. of Electronics,
[1]KIT's College of Engineering, Kolhapur, India

***Abstract* - The presented work proposes an algorithm that translates an image text from source language to destination language along with audio as well as visual output. A small scale demonstration has been done at the initial stage. The total concept has been processed in single software MATLAB. An algorithm initiates with an image processing technique called Segmentation followed by another one called Local Tetra Pattern. The source language i.e. a language of text present in an input image is a Marathi written in Devnagari script and destination language is English. Output is displayed with the help of MS Word application and audio version is provided through MS Excel. Input as well as output text is displayed and audio of translated output is given. We have got 100% result of both segmentation and character recognition for images considered.**

***Index Terms* - Content Based Image Retrieval (CBIR), Local Binary Pattern (LBP), Local Derivative Pattern (LDP), Local Ternary Pattern (LTP), Local Tetra Pattern (LTrP)**

I. INTRODUCTION

World is very rich in the field of linguistics. A mankind is blessed with thousands of languages [1], [2]. Though human beings are most intelligent among all the living beings it is next to impossible to have a thorough knowledge of each and every language. In this scenario it becomes essential to deal with this language barrier up to at least some extent. A mobile application can also be developed for this purpose [4]. The recent era which is developing day by day is a Technology. Utilization of this rapid growth can be done to find out solution of the mentioned scenario. A paper discusses a proposed work that may be a small solution. A presented algorithm initiates with the segmentation process. It is the process that divides input image into its constituent parts. After getting a desired output of segmentation the next part is to use tetra pattern technique to determine the image contents. Various ways of text recognition are given in [9], [10]. After determination of contents of input image those are displayed along with its translated version and also the audio of translated output.

2. Actual Work

A proposed algorithm deals with image of a text in specific font. Here Kruti Dev060 font is used. A specific font has been considered to simplify the initial level demonstration. A scope of work can be made wider in a future work. An algorithm uses MATLAB for execution. Fig 1 illustrates general idea of proposed work. An input image containing a sentence written in Devnagari script is taken. It then undergoes segmentation process which is subdivided as line segmentation, word segmentation, headerline removal and character segmentation. Result of this gives an individual image i.e. query image of all the characters present in sentence from input image. Further, different patterns are computed of both query as well as database images. These patterns are compared to recognize the character. Based on this comparison the recognized characters and sentence in input image as well as its translated version is displayed with MS Word application and the audio version of translated output sentence is provided through MS Excel.

A flow of work is explained below.

2.1 Thresholding

The work initiates with this image processing technique called Thresholding [3], [6], [7]. It is performed to convert an image into black and white. This determines a threshold value according to which pixels of image are categorized as black or white. A mathematical representation of it is given by (1).

$$g(x,y) = \begin{matrix} 1 & if\ f(x,y) > T \\ 0 & if\ f(x,y) \leq T \end{matrix} \qquad (1)$$

Here, T defines a threshold value which is constant and is applicable over an entire image. Eq (1) gives the process of global thresholding in which a single threshold value is considered.

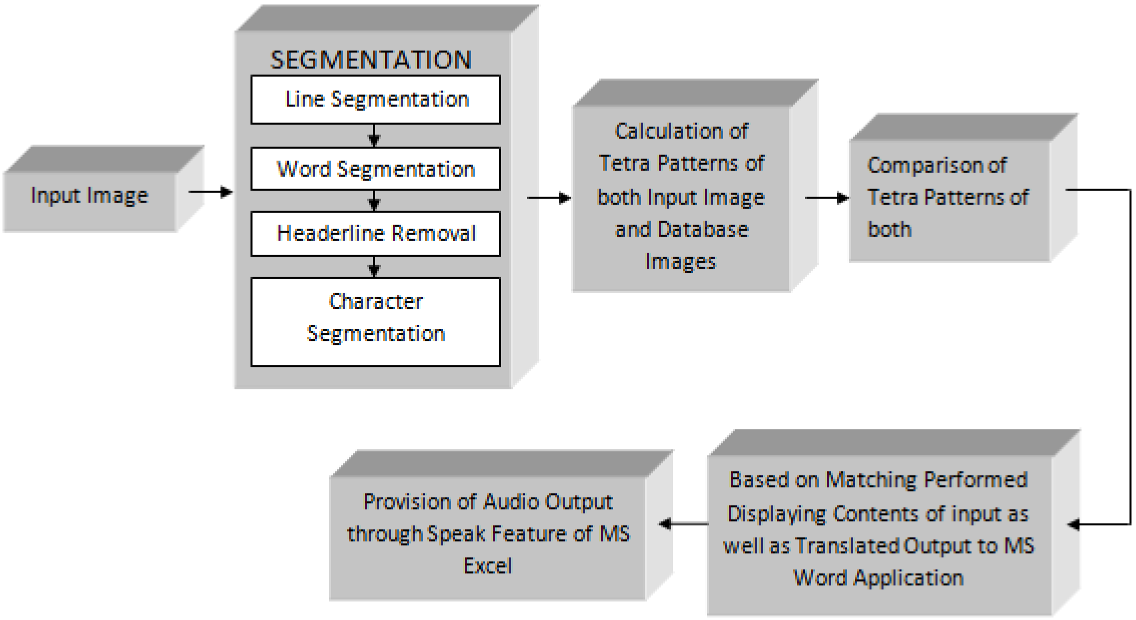

Input Image
SEGMENTATION
Line Segmentation
Word Segmentation
Headerline Removal
Character Segmentation
Calculation of Tetra Patterns of both Input Image and Database Images
Comparison of Tetra Patterns of both
Based on Matching Performed Displaying Contents of input as well as Translated Output to MS Word Application
Provision of Audio Output through Speak Feature of MS Excel

Fig 1: Generalized Bock Diagram

2.2 Segmentation

Once done with thresholding an input image undergoes segmentation. A work that integrates recognition and segmentation is given in [8]. Segmentation in our work has been done on the basis of discontinuity approach i.e. the abrupt change in intensity level has been a center part of logic. It is performed stepwise as given in [12]. Fig 2 illustrates flow of segmentation carried out. A sentence is segmented from the input image which is line segmentation and it is further segmented to get words. Headerline above word is removed to perform character segmentation. Separated characters are now ready to be identified. Similarly, Fig 3 describes another example.

2.2.1 Line Segmentation

A line from input image is segmented first on the basis of logic that a row without any black pixel is a separator between two lines. Algorithm searches for such two finitely distinct lines row and column wise. A part of an image present in between these two lines is nothing but a text in input image that is to be translated. Such segmented line image is shown in Fig 2 as well as in Fig 3.

2.2.2 Word Segmentation

Once done with the separation of line an input image is further scanned for column and row wise to find out the column with no any black pixel. Such a column is a separator between two words is the logic followed. The part of image between such two distinct columns is cropped and saved as a new image which is a image of word. Similarly, al the words from sentence are segmented. Result of this word segmentation is given in Fig 2 and Fig 3.

2.2.3 Headerline Removal

Headerline is a horizontal line drawn over words in Devnagari script. Now it is essential to remove this headerline present over the words for character segmentation. A logic used to detect such headerline present is to check for a row with maximum number of black pixels. Here some assumptions are made such as a headerline should have covered more than two rows and the number of black pixels in al headerlines should be equal or can have maximum difference of 6 pixels. A threshold of 6 pixels is considered by us. It can be changed according to requirement. A pixel is very tiny part of an image. Headerline is thick as compared its width with the size of pixel. These assumptions are made for exact detection of headerline from image. A detected headerline is removed just by inverting those black pixels of rows to white.

2.2.4 Character Segmentation

A similar logic as that of word segmentation is used here. A column with no any black pixel is a separator between characters.

All these segmentation results are given in Fig 2 and Fig 3 showing two different examples. Now, every character from input image is ready to be recognized.

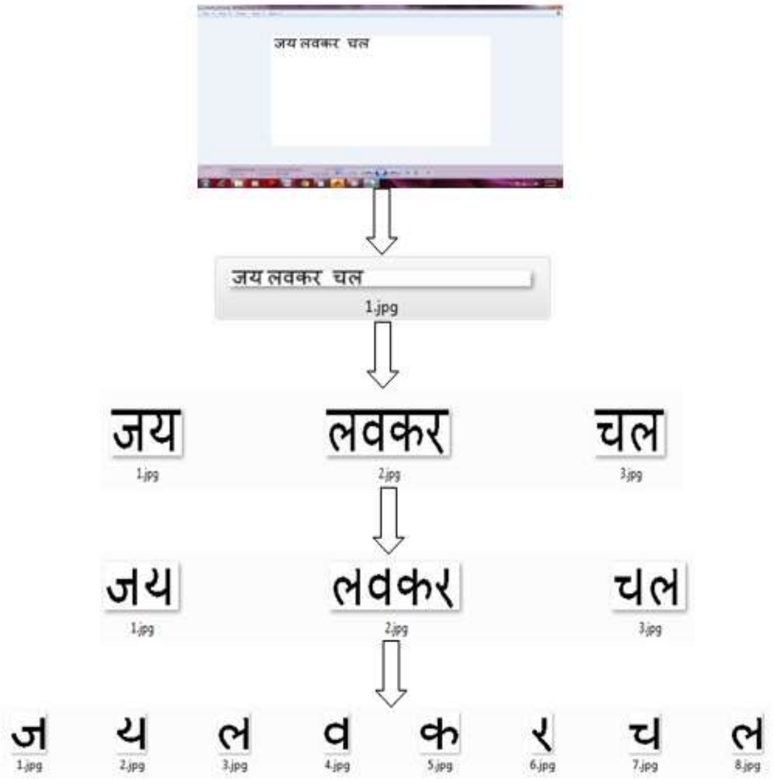


Fig 2: Segmentation of Image1

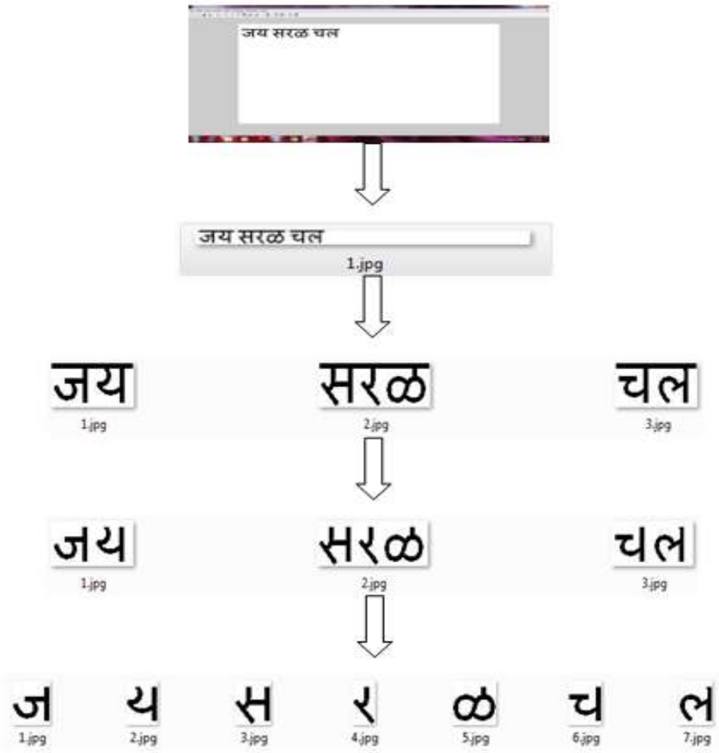


Fig 3: Segmentation of Image2

2.3 Local Patterns

Separated characters are to be translated into destination language. Hence those should be recognized first. There is a database formed which contains character images. Separated character image is matched with every image from database. This technique is known as Content Based Image Retrieval (CBIR). Different approaches of CBIR are given in [14]. A search is carried out on the basis of contents present in image such as colour, shapes, textures or any other information which can be retrieved from image itself. For this a technique used here is Local Tetra Pattern. There are some local patterns that are used in extraction of information from an image which is based on the distribution of edges. These are coded with two directions known as positive or negative direction.

2.3.1 Local Binary Pattern (LBP)

It is computed by comparing center pixel with its 8 neighbors on its left-top, left-middle, left-bottom, right-top likewise. The pixels are followed along the circle either in clockwise or anticlockwise direction. One more application of LBP is given in [16].

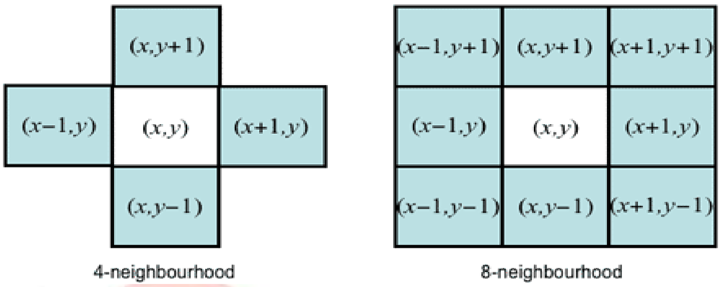


Fig 4: Neighborhood

$$f(p, g_c) = \begin{cases} 0 & \text{if } g_c > p \\ 1 & \text{else} \end{cases} \qquad (2)$$

Eq (2) gives mathematical representation of LBP. If the center pixel g_c has a value greater than the neighbor pixel value p then LBP is coded as 0 otherwise 1. Each value of 8 neighbors forms 8 bit LBP. Fig 5 illustrates LBP computation.

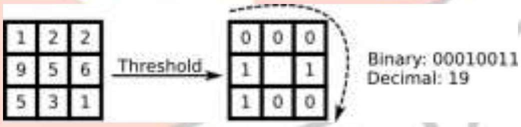


Fig 5: LBP

2.3.2 Local Ternary Pattern (LTP)

Unlike LBP it does not use 0 and 1. Rather it uses a threshold constant according to which pixels are thresholded to have three values.

$$f(p, g_c, t) = \begin{cases} 1 & \text{if } p \geq c + t \\ 0 & \text{if } p > g_c - t \text{ and } p < g_c + t \\ -1 & \text{if } p \leq g_c - t \end{cases} \qquad (3)$$

Here, t is a threshold constant, g_c is a center pixel value and p is a neighboring pixel value. Likewise each pixel has three values. Fig 6 gives LTP computation.

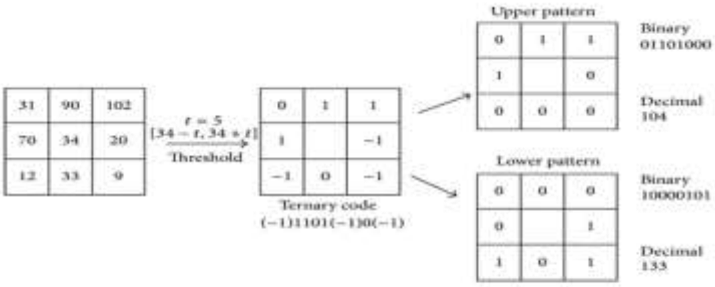
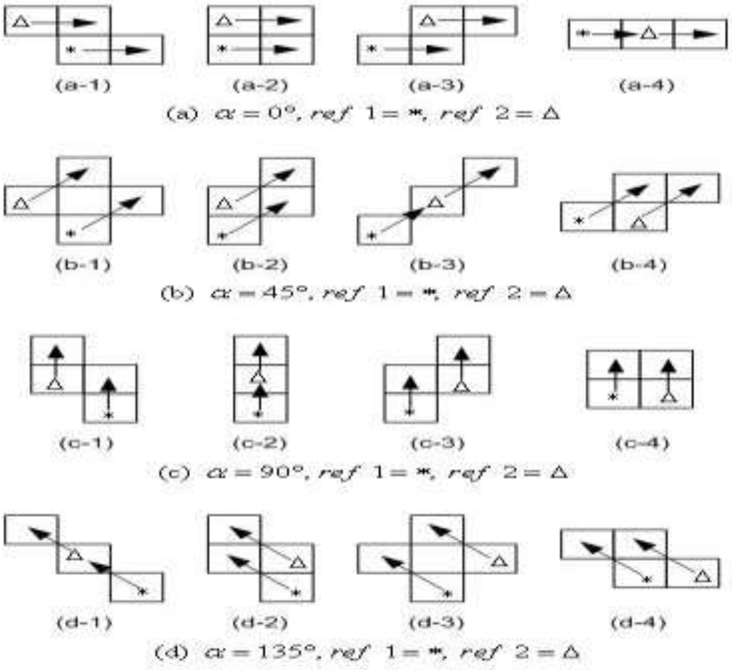


Fig 6: LTP

2.3.3 Local Derivative Pattern (LDP)

It encodes (n-1)th order derivative direction variations. These are on the basis of binary coding function. Result of LDP contains more detailed features since it encodes the higher-order derivative information. The nth order LDP is determined by calculating (n-1)th order derivative along the directions 0°, 45°, 90°, 135°. Fig 7 illustrates computation of LDP.



(a) $\alpha = 0°$, $ref\ 1 = *$, $ref\ 2 = \Delta$

(b) $\alpha = 45°$, $ref\ 1 = *$, $ref\ 2 = \Delta$

(c) $\alpha = 90°$, $ref\ 1 = *$, $ref\ 2 = \Delta$

(d) $\alpha = 135°$, $ref\ 1 = *$, $ref\ 2 = \Delta$

Fig 7: LDP

2.3.4 Local Tetra Pattern (LTrP)

A description about how objects of texture are fit together is given by LTrP. Basic relationship between center pixel and its neighborhood is encoded by LTrP. It uses a direction of center pixel g_c. LTrP is a feature descriptor in CBIR [18]. An image I contain multiple pixels among which for any center pixel g_c has 8 neighborhood as shown in Fig 4. Among those two fix neighbors i.e. horizontal (g_h) and vertical (g_v) are considered. These are on directions 0° and 90° respectively. These directions and neighbors are shown in Fig 8. Then the first-order derivatives along these directions are given by

$$I_{0^o} = I(g_h) - I(g_c) \tag{4}$$

$$I_{90^o} = I(g_v) - I(g_c) \tag{5}$$

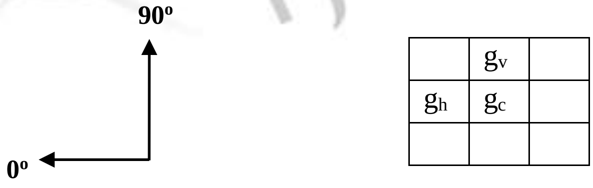


Fig 8: Directions and Neighborhood

As like horizontal and vertical there are total 4 possible directions of a center pixel given as

$$I_{Dir}(g_c) = \begin{cases} 1, & I_{0^o}(g_c) \geq 0 \text{ and } I_{90^o}(g_c) \geq 0 \\ 2, & I_{0^o}(g_c) < 0 \text{ and } I_{90^o}(g_c) \geq 0 \\ 3, & I_{0^o}(g_c) < 0 \text{ and } I_{90^o}(g_c) < 0 \\ 4, & I_{0^o}(g_c) \geq 0 \text{ and } I_{90^o}(g_c) < 0 \end{cases} \tag{6}$$

LTrPs of higher-order can also be computed. Fig 9 and Fig 10 illustrates how LTrPs are determined. Fig 9 shows that for a center pixel with direction 1 and respective directions of neighbors what will be the direction pattern. Basically a Tetra Pattern is a pattern calculated according to directions of center and neighbors.

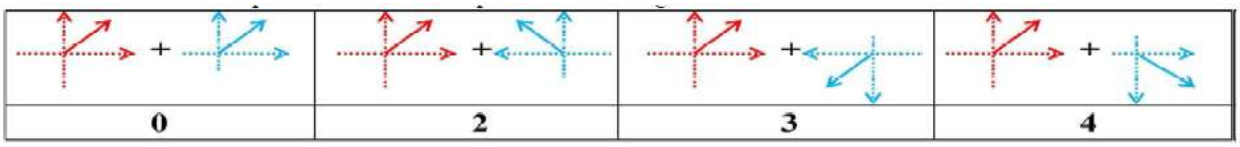


Fig 9: Calculation of Tetra Pattern for Direction 1 of center pixel using direction of neighbors

Initially, the direction of center pixel is calculated. Every pixel has its 8 neighbors which are called as reference pixels. Consider example in Fig 10 in which 4 is a center pixel and 8 is a reference pixel which is horizontal neighbor of center pixel while 9 is a vertical neighbor. Here both the neighbors are having greater value as that of center pixel hence direction of center pixel is Direction 1. Now, directions of its 8 neighbors are calculated one by one. These values form an 8 bit tetra pattern. Every bit is decided by comparing directions of center and its neighbor. If both center and a neighbor have exact same direction then pattern bit is coded as “0” otherwise it is coded as the value of direction of neighbor pixel. Fig 9 explains this in a better manner. First 0 shows that directions of both center and neighbor are same hence it is coded as zero. In next case direction of center is 1 but direction of neighbor is 2 hence bit pattern is coded as the direction same as that of neighbor i.e. 2. With this an bit tetra pattern formed is “30340320” as shown in Fig 10.

Another part to be performed is to compute magnitude which is given by

$$\left|g_c\right| = [(\text{Horizontal Neighbor} - \text{Center Pixel})^2 + (\text{Vertical Neighbor} - \text{Center Pixel})^2]^{1/2} \qquad (7)$$

$$\left|g_{ref}\right| = [(\text{Horizontal Neighbor} - \text{Center Pixel})^2 + (\text{Vertical Neighbor} - \text{Center Pixel})^2]^{1/2} \qquad (8)$$

From this a magnitude pattern is generated. It is coded with logic that if magnitude of center pixel is greater than magnitude of reference pixel then a pattern bit is assigned value “0” otherwise “1”. Fig 10 shows a magnitude pattern of “11100101” for that example.

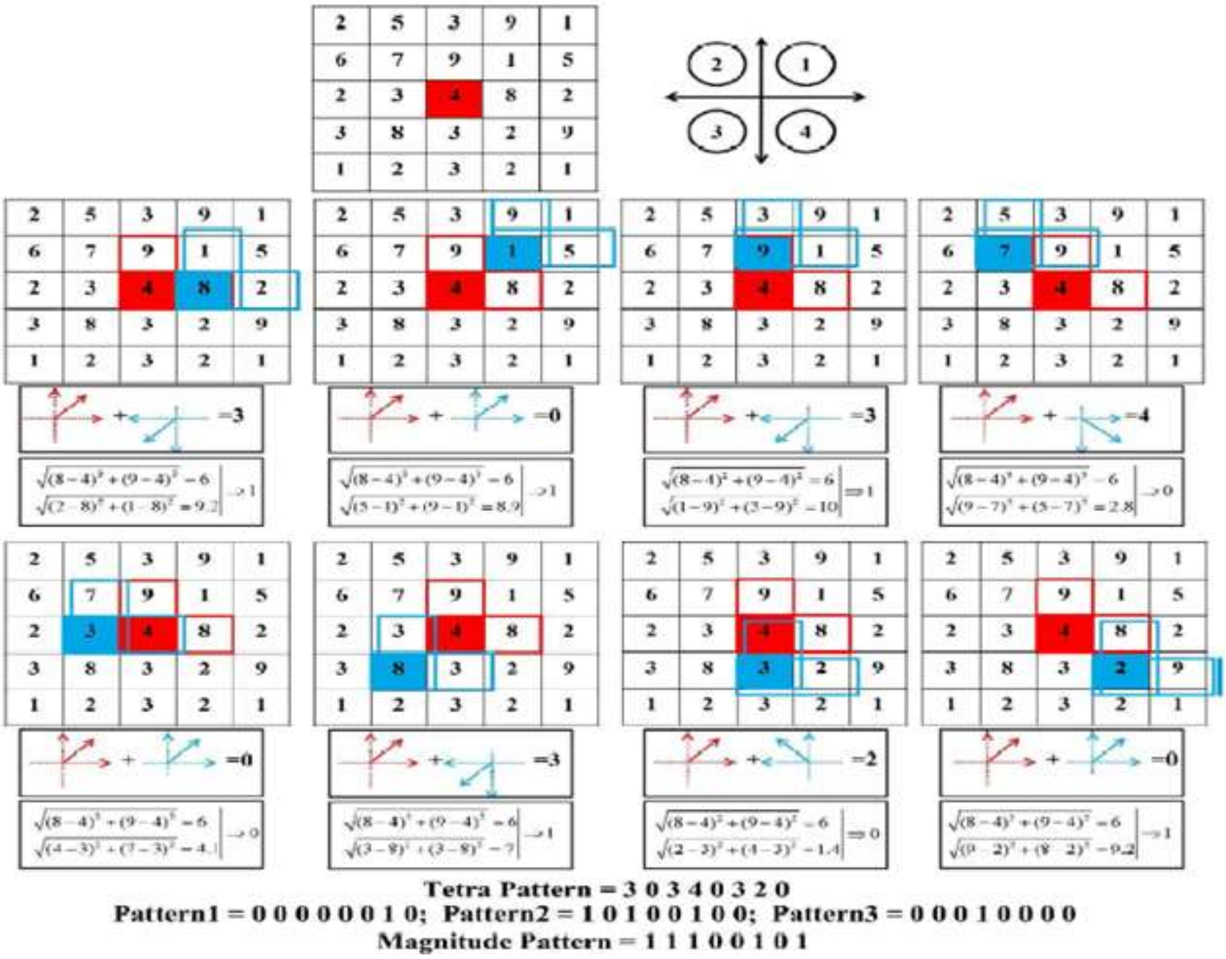

Tetra Pattern = 3 0 3 4 0 3 2 0
Pattern1 = 0 0 0 0 0 0 1 0; Pattern2 = 1 0 1 0 0 1 0 0; Pattern3 = 0 0 0 1 0 0 0 0
Magnitude Pattern = 1 1 1 0 0 1 0 1

Fig 10: Example showing computation of Magnitude and Tetra Pattern

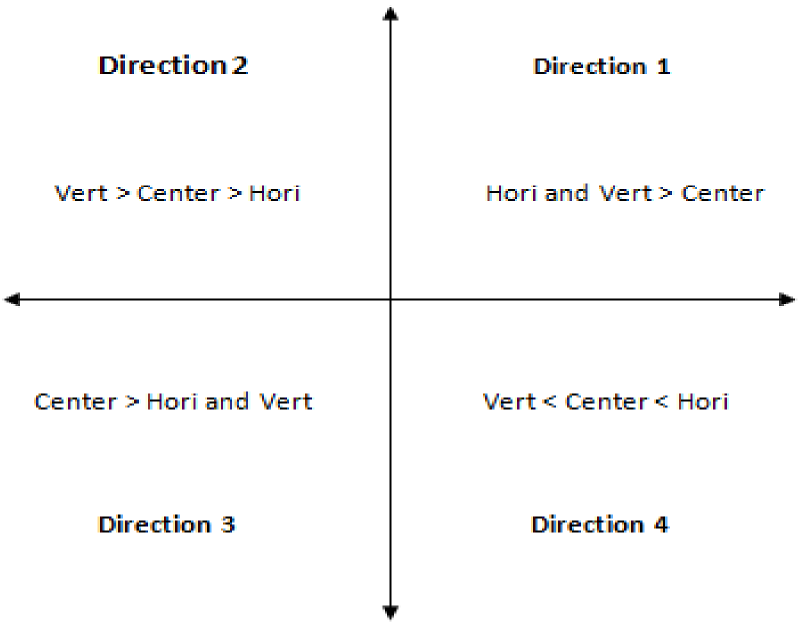


Fig 11: Directions according to relationship between Center pixel and its neighbors

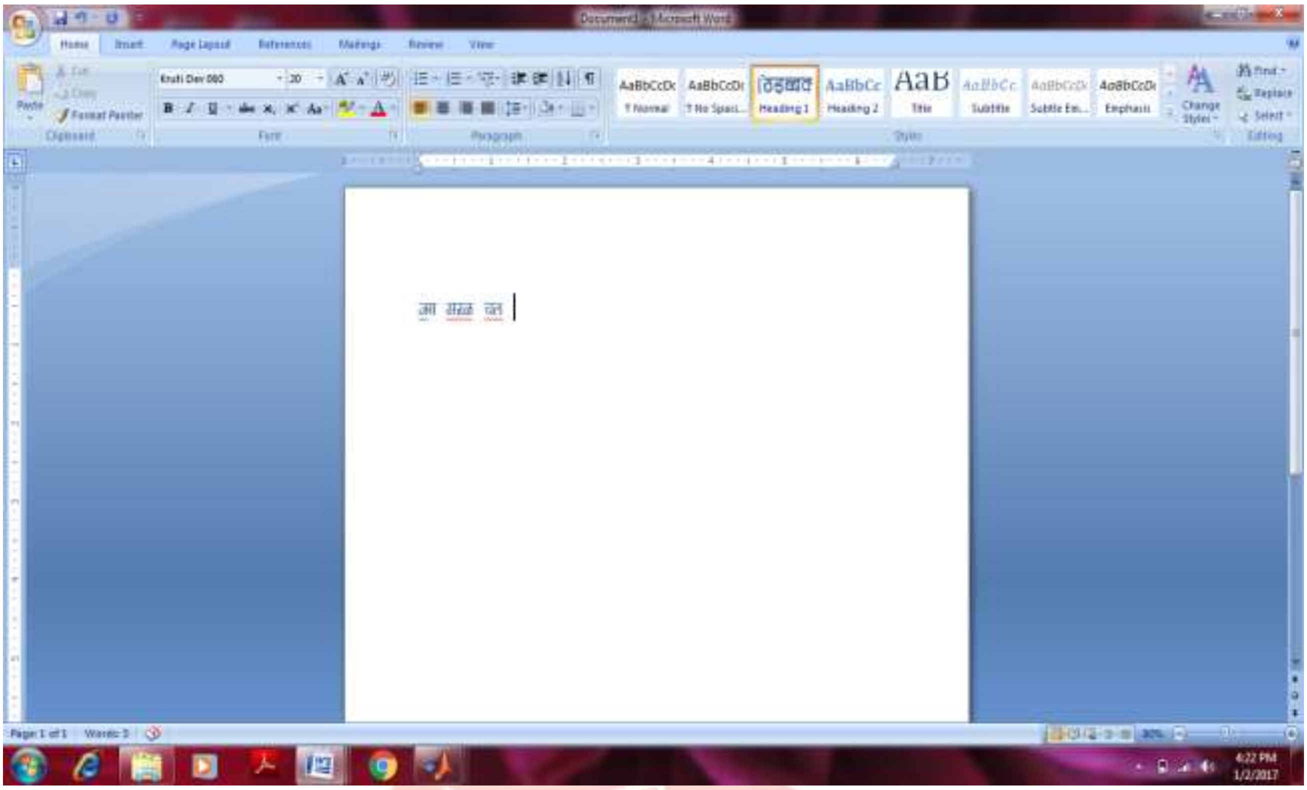
Once tetra pattern is coded it is further separated into four binary patterns i.e. each tetra pattern forms four binary patterns. The first one is Zero Pattern which is formed by replacing 0 by 1 in a tetra pattern rest of the bits are coded as 0. Next, Two Pattern is calculated by replacing 2 by 1 and rest bits by 0. Similarly, Three Pattern and Four Pattern are calculated by replacing 3 by 1 and 4 by 1 respectively keeping other bits 0 of tetra Pattern. In similar way tetra patterns of center pixel in direction 2, 3, 4 are determined along with their respective binary patterns. Hence, total 17 binary patterns are computed for each center pixel. Four tetra patterns each having 4 binary patterns equals 16 patterns and 17th one is a magnitude pattern.

In our proposed work such patterns are calculated for all segmented character images and also for the images stored in database. Input images (e.g. Image1 and Image2) contains single sentence written in Devnagari script which forms 7 and 8 constituent character images respectively representing every character from sentences. As it is an initial level demonstration of the concept database contains only those character images which are present in input images. In case of Image1 as it splits into seven character images (query images) each character image will be having 16 binary patterns and a magnitude pattern for every center pixel. Each of the character images from database will also be having such 17 binary patterns (4 binary patterns for each direction equals 16 patterns and a magnitude pattern). While determining a first character from input image all these patterns of every pixel will be matched i.e. patterns of query image and database image. When all the patterns of query image matches with all patterns of particular database image then that character is identified. Likewise, every query image contents are determined. This much big computations increase operation time. A solution for this is to resize the images which are to be matched to smaller size. Here we have resized to 15X15. The drawback of this is reduction in efficiency. Hence have to make a tradeoff between efficiency and operation time. We have got the perfect results despite of it. Recognized Devnagari characters are displayed to MS Word Application one by one to form a complete sentence present in input image. Based on recognized characters the translated version of input sentence is also displayed with MS Word Application with the help of ASCII values. Further, speak support is provided by utilizing speak feature of MS Excel and it gives audio version of translated output text. All these operations are performed in MATLAB itself.

3. Conclusion

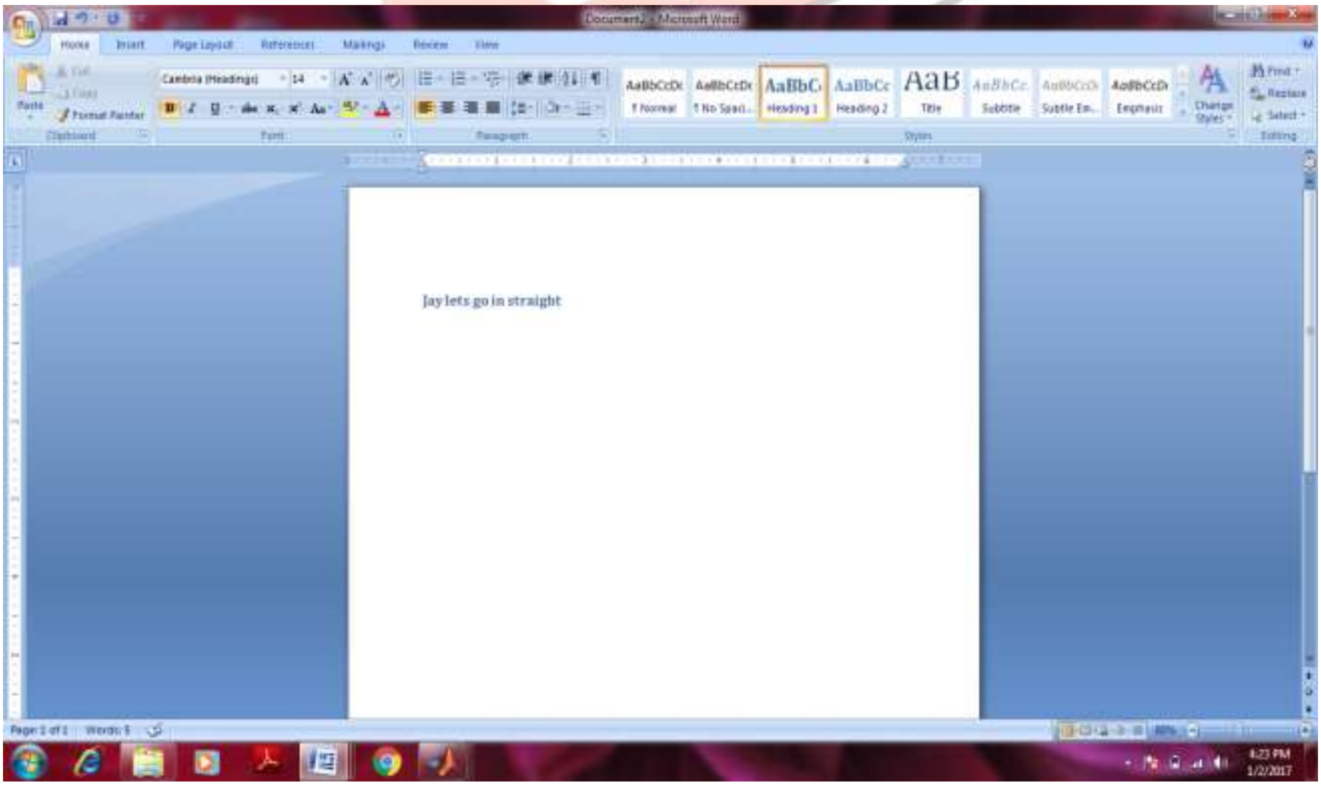
The presented work in this paper takes the contents of input image which is a sentence written in Devnagari script and provides audio as well as visual output of its translated version. Images with specific font i.e. Kruti Dev060, required database and Devanagari sentences with no any special symbols like ‘Kana’, ‘Matra’, ‘Velanti’, coma, full stop etc are considered because it is an initial demonstration done here. LTrP has been used for character recognition which is of low accuracy but output is unaffected despite of it. Multiple images having different sentences with specific font have been processed. Fig 12 and Fig 13 shows two examples of received results i.e. displayed contents of input image1and input image2 respectively. A proposed algorithm gives a perfect result of 100% for considered images in case of both segmentation and character recognition.

4. Results



जा सरळ चल

Fig 12: Displayed Contents of Input Image



Jay lets go in straight

Fig 13: Displayed Translated Output

5. Future Work

Further an algorithm can be expanded to process big contents such as paragraphs written in Devnagari script. A database needs to be updated accordingly. Intelligence about special symbols can be added. An accuracy of tetra pattern technique may be improved but it will slower the operation time which may not be desirable.

6. Acknowledgement

I take this opportunity to be thankful of all the teachers, HOD and non teaching staff of our college who have helped us. Special thanks to co-author for his support and help. I am grateful to everyone helped us to bring this work to success.

REFERENCE

[1] https://www.google.co.in/url?sa=t&source=web&rct=j&url=http://www.andamanese.net/Languages%2520of%2520india%2520and%2520india%2520as%2520a%2520linguistic%2520area.pdf&ved=0ahUKEwj6rsi8oKrRAhWIKo8KHc70CWwQFggjMAM&usg=AFQjCNFn-kWsqztZ7THWJMWdq87LBIgpEQ

[2] https://www.google.co.in?url?sa=t&source=web&rct=j&url=http://tdil.mit.gov.in/wsi/papers/511432Development_of_Linguistic_Resources_and_Tools_for_providing_multilingual_.pdf&ved=0ahUKEwj6rsioKrRAhWIKo8KHc70CWwQFggpMAU&usg=AFQjCNEYQpl0qjr76VqOK3hMG_n5k6MmEQ

[3] Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing" Third Edition, Pearson

[4] Edgard Chammas, Chafic Mokbel, Rami Al Hajj Mohamad, Christina Oprean, Laurence Likforman Sulem and Gérard Chollet, "Reducing Language Barriers for Tourist Using Handwriting Recognition Enabled Mobile Application", 2nd *International Conference on Advances in Computational Tools for Engineering Applications* 2012

[5] A. A. Tayade, Prof. R. V. Mante and Dr. P. N. Chatur, "Text Recognition and Translation Application for Smartphones", *International Journal of Advanced Research in computer and Communication Engineering Volume 2 Issue 6*, November 2013

[6] Salem Saleh Al-Amri, N. V. Kalyankar, S.D. Khamitkar, "Image Segmentation by Using Threshold Techniques", *Journel of Computing, Volume 2, Issue 5*, May 2010

[7] http://www.math.tau.ac.il/~turkel/notes/threshold.pdf

[8] Jerod J. Weinman, Zachary Butler, Durgan Knoll, Jacqueline Field, "Towards Integrated Scene Text Reading", *IEEE Transaction on Pattern Analysis and Machine Intellegence, Volume 36, No 2*, February 2014

[9] Q. F. Wang, F. Yin,C. L. Liu, "Handwritten Chinese Text Recognition by Integrated Multiple Contexts", *IEEE Transaction on Pattern Anaysis and Machine Intelligence, Volume 34, No. 8*, August 2012

[10] L. Neumann, J. Matas, "A Method for Text Localization and Recognition in Real-World Images", *Proc. Asian Conference on Computer Vision*, 2010

[11]Brijmohan Singh, Nitin Gupta, Rashi, Debashish Ghosh, " Parellel Implementation of Devanagari text Line and Word Segmentation Approach on GPU", *International Journal of Computer Applications, Volume 24, No. 9*, June 2011

[12] Pulkit Goyal, Sapan Diwakar, Anupam Agrawal, "Devanagari Character Recognition Towards Natural Human-Computer Interaction" *Proceedings of India HCI*, 2010

[13] Bikas Shaw, Swapan Kr. Parui, Malayappan Shridhar, "Offline Handwritten devanagari Word Recognition: A Segmentation Based Approach" *IEEE*, 2008

[14] Jagpal Singh, Jashanbir singh kaleka, "Different Approaches of CBIR Techniques", *International Journal of computers & Distributed Systems, Volume 1, Issue 2*, August 2012

[15] Y. Liu, D. Zhang, G. Lu and W-Y. Ma, "A Survey of Content- Based Image Retrieval with High-level Semantics", *Pattern Recognition*, Volume 40, No. 1, January 2007

[16] T. Ojala, M. Pietikainen, T. Maenpaa, "Multiresolution Grey-Scale and Rotation Invarient Texture Classification with Local Binary Patterns", *IEEE Transaction on Pattern Analysis Mach. Intell., Volume 24 No. 7*, July 2002

[17] S Anitha, A. Jeeva, Niveditha R. Das, K. Yoheswari, P. Devi, "Content Based Image Retrieval Using Local Tetra Pattern", *International Journal of Electronics, Communication & Instrumentation Engineering Research and Development (IJECIERD) Volume 3,Issue 1*, March 2013

[18] Subrahmanyam Murala, R.P. Maheshwari, R. Balasubramanian, "Local Tetra Patterns: A New Feature Descriptor for Content-Based Image Retrieval" *IEEE Transaction on Image Processing, Volume 21, No.* , May 2012