This full-text paper was peer-reviewed and accepted to be presented at the IEEE ICCSP 2015 conference.

Hindi Speech Synthesis by concatenation of recognized Hand written Devnagri Script using Support Vector Machines Classifier

Saurabh Farkya, Govinda Surampudi, and Ashwin Kothari

Abstract— **Handwritten optical character recognition is one of the major research area due to its complexity in segmenting the character which increases in the case of Devnagri Script due to Modifiers and compound characters. Thus this paper shows an adaptive segmentation technique which shows less error. This paper shows an implementation of Handwritten Devanagri character recognition system. Keeping in mind the impairment of blind people, OCR system is extended to Text to Speech System. As by the surveys Support Vector Machines found to be very efficient and robust in handling large amount of features, SVM was used for the purpose of classification. The learning was done in Multi domain feature. In Transform domain, wavelet transform was used because of its ability to keep global feature in different scale. In structural domain, gradient and distance profile features were used as they represent local characteristics of the characters. Here the handwritten document was segmented adaptively in 3 levels; line, word and character. Preprocessing was done using various Morphological operation and thinning techniques. All the training was done using self-created database consists of 50 samples per character from 10 different individuals. Further, the recognized characters were digitized using Unicode and their corresponding phoneme present in the database were concatenated to form the speech signal. A self-created database of all the possible phonemes was created.**

Keywords— **Optical Character Recognition (OCR), Text to Speech (TTS), Support Vector Machines (SVM), Handwritten Optical Character Recognition (HOCR), LIBSVM(Library of Support Vector machines).**

I. Introduction

OCR and TTS of Devnagri Script are the two types of system designed in this paper. Both the system have huge application for practical purposes such as information retrieval from rich ancient Indic Text and as an assistance to blind. TTS system consists of speech synthesizer to generate artificial voice.

Saurabh Farkya is with the Dept. of ECE VNIT Nagpur, India.
Govinda Surampudi Farkya is with the Dept. of ECE VNIT Nagpur, India.
Ashwin Kothari is with the Dept. of ECE VNIT Nagpur, India.

Many methods are used to implement TTS such as Concatenation, Formant, Articulatory, HMM based etc. Here Concatenation method was applied. However the efficiency of TTS directly depends up on that of OCR. Thus a highly accurate and robust OCR system is required. A well optimized HOCR can predict handwriting written by a person of any age, gender, educational background as well as in any mood. The accuracy of OCR depends directly on the segmentation methods and the features used for learning. Here, due to adaptive nature of the segmentation OCR automatically adjusts according to the modifiers and character's sizes. Segmentation was done in 3 levels; line, word and character segmentation. Before the segmentation, the characters were individually preprocessed using various Morphological operations and filtering to reduce them to one pixel thin image to extract feature. Feature used here capture both global and local feature of the character. Transform Domain maps the Global features of the characters by wavelet transform in coefficients. Pixel values of the character image become a function of wavelets whose coefficients are used as one of the features. Wavelets are functions able to capture frequency and positional information of the image. Spatial Domain technique extracts features from the structural pattern of pixel representation. Depth or distance of the outer strokes from specific directions are measured by distance profile. Gradient features measure the rate of change of curvature of the strokes. The final feature vector is a concatenation of the three features.

The other factor on which the efficiency and speed of OCR depends up on the learning Mechanism used. Learning theory's main purpose is to develop an approximate function from a set of hypothesis that nearly equates to the original unknown function. Here the learning mechanism used was Support Vector Machines which is based on the statistical learning theory. SVM is a binary classification algorithm but it has been extended to an application of multi-class classification of characters due to its ability to generalize and generate classifiers that can handle large amount of features and classes.

978-1-4799-8081-9/15/$31.00 © 2015 IEEE

Support Vector Machines are mechanically motivated theory and can be viewed as, a plane sheet between two types of particles, which exert force in opposite directions, trying to balance itself. N classifiers are placed in parallel and each class is trained against the rest. Further, the recognized text is converted to digital text using Unicode scheme. Finally, Hindi digital Text is generated in a text file, which is an array of generated Unicode used to synthesize Hindi speech. Devnagri script comprises of 33 consonants and 14 vowels. The characters are combined in only these five ways- V, C, CV, CCV, CCCV; C- consonant, V-vowel. Thus using the method of concatenation we generate speech using the pre stored phonemes in the database of Devnagri characters. This paper is organized as follows. Section II shows related work of all the algorithms and methodology, section III elaborates basic characteristics of the Devnagri script, section IV describes about the created database, section V shows a flowchart of the whole process, section VI shows pre-processing methods, next section shows an implementation of adaptive segmentation. Section VIII shows all the feature extraction methods, section IX describes the learning theory for the classification, section X describes the methodology for speech synthesis. Finally reference section follows the conclusion and result section.

II. Related work

The research and Developments on OCR and TTS is going throughout the world in various languages. Many researchers are separately working on the challenges of such systems to make OCR system more robust and to make TTS system generate human like voice. One such initiative is taken by Indian govt. which motivated us to pursue our research in one the most ancient script – Devnagri Script. The Programme initiated by the Department of Electronics & Information Technology (DeitY), Ministry of Communication & Information Technology (MC&IT) under the name Technology Development for Indian Languages (TDIL)[1]. This Programme put efforts towards development of multilingual knowledge resources. Until now TDIL has development many products such as Anuvadaksh, Anglabharti, and Sampark etc. Many prestigious institute of India are working for this project headed by IIT Madras.

The various stages of OCR and TTS system in this paper follows the below given research work. Most of the pre-processing techniques are directly developed in various libraries of Matlab and OpenCV in the form of functions. For segmentation Sirisha Badhika [2] and Ashwin S Ramteke et.al [3] have worked on 3 level segmentation which in this paper has been modified to adaptive 3 level segmentation to reduce error. For feature extraction many research works have been referenced to cover wide variety of work. Referred Ovinind et.al [4] work shows survey of structural features. Based on various Indian languages U. Pal and B.B. Choudhary [5] discussed a survey many feature and their possibilities for future. An implementation of wavelet feature and gradient feature is shown by Weipeng Zhang et.al [6] on Chinese characters. Also the book by Rafael Gonzalez [7] helps us in understanding the fundamentals of wavelet transform. Coming to learning theory, Chih-Wei et.al [8] provides brief detail about the practical implementation of SVM. Also the work of Christopher J.C. Burges [9] helps in understanding the mechanical interpretation of the SVM. However, a lot help was provided by the FAQ section of LIBSVM library [10] which was used for the implementation of this paper. For TTS, Hitachi Ltd. Central research Lab. [11] work was followed which shows a TTS based on scalable implementation of unit selection. Also K. Pratha Sarathy et.al [12] was referred, who developed Tamil TTS for embedded application. People from IIIT-H, IIT-K have also worked on natural sounding speech synthesize of Indian Languages.

III. Characteristics of Devnagri script

Devnagri script is the primitive language based on which many Indian languages are developed such as Gujrati, Marathi etc. There are in total 33 consonants and 14 vowels in the script. An independent vowel or a consonant vowel combination is termed as a syllable. Devnagri script defers to syllabary, i.e. syllables representing distinct phonemes; or graphemes or syllable being the same. Thus a grapheme in this script is a vowel, or a consonant vowel combination. The specialty of Devnagri script is that graphemes and phonemes have a one-to-one correspondence. Consonants cannot be pronounced directly without any vowels. Consonants usually carry an inherent vowel, first vowel, but can be modified diacritically to carry other vowels. However, vowels can be pronounced independently.

Vowels can be written independently or can be incorporated in consonants by placing diacritic marks before, after, below, or above consonant symbol. Modifiers are used for combining vowels and consonants. Consonants can be combined to form conjuncts as mentioned before- C, CC, CCC, with a vowel based on principles on phonetic articulation.

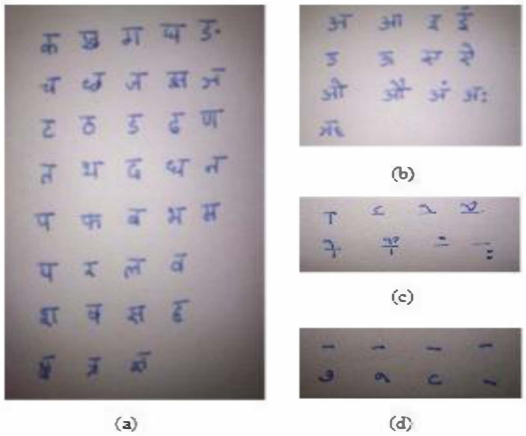

Fig. 1. (a) Consonants. (b) Vowels. (c) Upper Modifiers. (d) Lower Modifiers

IV. DATABASE

A. Database for Training OCR

For training, a self-created database was used. Database consists of 74 consonants and vowels. Each character has 50 samples taken from handwritings of 10 different persons. Total of 3700 characters to train. Also, 5 upper modifiers and 2 lower modifier each with 15 different samples were used for training.

B. Database for Speech Synthesis

For mapping the recognized text with the corresponding phoneme, all possible consonant vowel combinations were recorded clearly and stored in the database. Totally 422 .wav files were stored in the database.

V. FLOWCHART

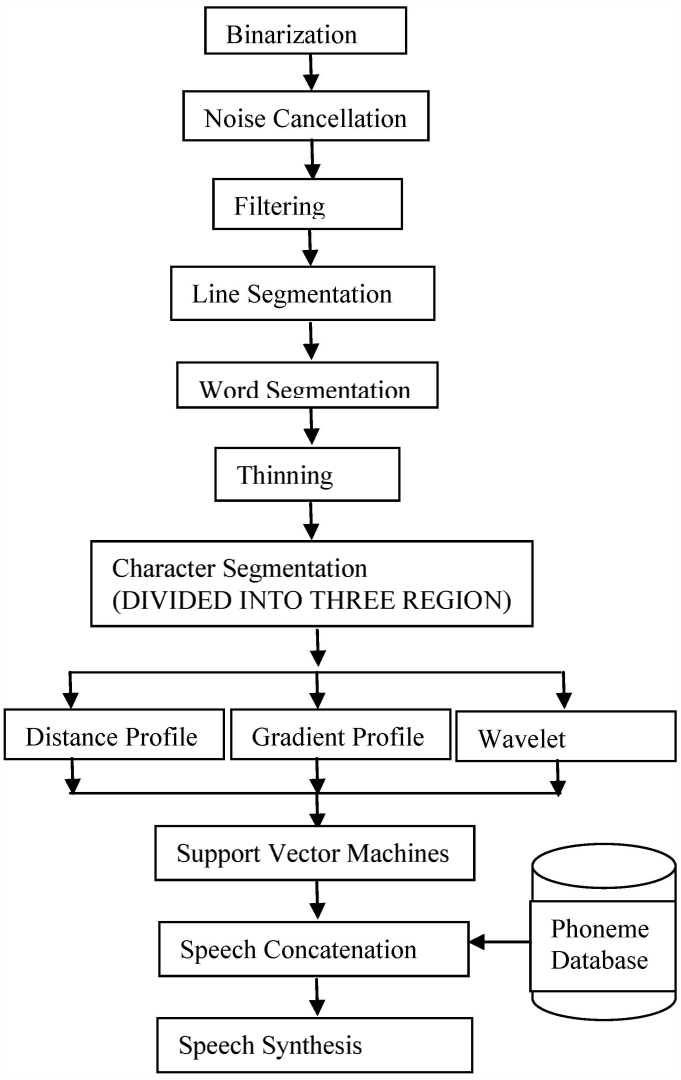

Fig. 2. Flowchart of OCR and TTS.

This Flowchart describes all the stages of the OCR and TTS as shown in Fig. 2.

VI. PRE-PROCESSING

Pre-processing is the technique of emphasizing data in image and deemphasizing noise from image prior to computational processing. The scanned image is rgb in nature, hence before converting it into a binary image it has to be converted to grayscale as shown in Fig. 3 and 4.

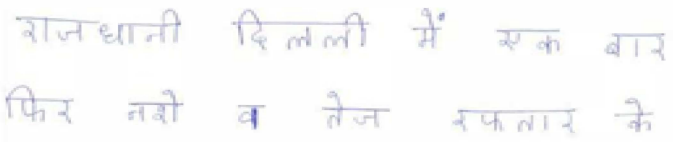

Fig. 3. Scanned Image

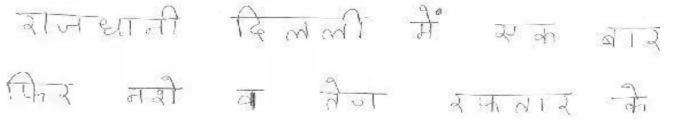

Fig. 4. Grayscale Image

A. Binarization

It is process of converting a scanned image into binary image. A binary image comprises all its pixel value as either1 or 0 using threshold obtained from Otsu's Algorithm.

B. Noise Removal

The unwanted group of pixel values were removed from the image. Noise up to 5px or salt noise was removed.

C. Filtering

Filtering is a technique for modifying or enhancing an image, and to remove background noise. It involves neighborhood operations, in which the value of any given output pixel value is determined by convolving the filter mask with neighborhood pixels. But here, median filtering technique was used. It is a non-linear operation used to remove salt pepper noise. It is more effective than convolution because it preserves edges and reduces noise as shown in Fig. 5.

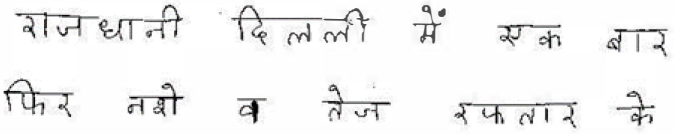

Fig. 5. Filtered and Binarised Image

VII. ADAPTIVE SEGMENTATION

A. Line Segmentation

Individual lines of scanned image were cut using the horizontal projection method, a horizontal histogram of white pixels along every image row was calculated and by keeping a certain threshold (as per error) lines were segmented out as shown in Fig. 6.

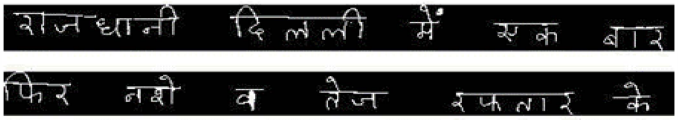

Fig. 6. Lines are segmented from the document

Steps for Calculation of average height of character:
• Height of all the lines obtained from the line segmentation were calculated.
• Median of all those lines was obtained.
• In this step, by analysis it was assumed that the consonant height takes up to 30% of the height of average height of the strip (It may vary according to the database used)
• Hence height of character was obtained.

B. Word Segmentation

From the stripes obtained, words using Vertical projection method in view of shirorekha were extracted. Here, the value of threshold used here was up to 4 pixels (as per error) in view of extended shirorekha as shown in Fig. 7.

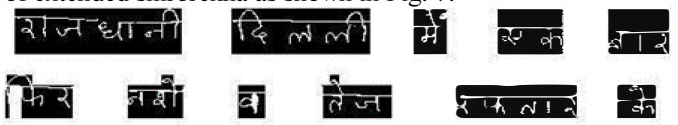

Fig. 7. Words are segmented from each Line

C. Thinning

Before character segmentation, the words extracted out were skeletonized. Thinning comes under morphological operations that selectively removes boundary or foreground pixels to make the black and white image data one pixel thin stroke as shown in Fig. 8.

Fig. 8. Each Character is thinned

D. Character Segmentation

In order to separate matras or Upper modifiers from the characters, upper half of the word stripes were scanned for the highest value (or largest flat portion) of the horizontal histogram in search of the header line. Approximately twice the width of the matra was then reserved for the consonant or middle stripe and same size as of upper modifier was taken for lower modifier and hence the three sections were obtained. Total width was then close to four times the matra width.

But if the upper modifier was not present then the height of consonant was taken as 1.5 times the size of consonant obtained earlier and the below modifier was calculated approximately from the formula (Total size of character – height of the character). Hence, character and below modifier were obtained.

These obtained stripes were slightly overlapped by 5 pixels from each intersecting stripe, i.e. three stripes obtained were overlapped up to 10px in order to correctly obtain the required features per character.

VIII. FEATURE EXTRACTION

Feature extraction is a methodology used to find characteristics of pattern in the subject of our application.

A. Distance Profile Feature

The distance of outer edge of the character from a specific direction was measured as shown in Fig. 9(a-d) . Here left, right, top, bottom viewed distances were measured and these features were concatenated to form a single feature vector. Size of each training character was taken to 45*45 px image. Below figures show profile of letter 'ta'.

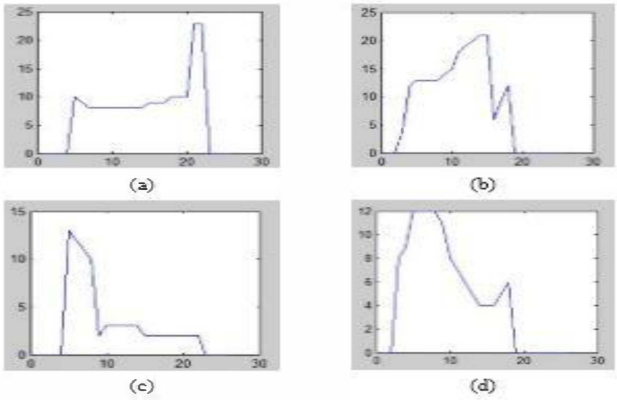

Fig. 9. (a). Bottom profile. (b) Right profile (c) Top Profile (d) Left profile.

B. Gradient Feature.

Gradient Feature measures the directionality of edges in the feature space within the bounding box. It is visually a very prominent characteristic and highly stable for grouping various shapes. Here, extraction of directional features was via the directional decomposition of gradient map. The gradient direction features were obtained in three steps: gradient computation, directional decomposition, and feature reduction. The binary image at each stroke edge was convolved with two Sobel operators to obtain the decomposed horizontal gradients and vertical gradients. The gradient vector has both direction and magnitude, so only direction was used for feature computation. Each vector ranges an angle from 0 to 359 degrees. This range is split into 12 chaincode directions of 360/12 degrees apart

Below are the equation 1 gradient and eq. 2 for magnitude and are taken from the work done in the paper [6]:

$$f_v(i,j)=g(i-1,j+1)+2g(i,j+1)+g(i+1,j+1)-g(i-1,j-1)-2g(i,j-1)-g(i+1,j-1)$$
$$f_h(i,j)=g(i-1,j-1)+2g(i-1,j)+g(i-1,j+1)-g(i+1,j-1)-2g(i+1,j)-g(i+1,j+1)$$

$$F(i,j)=\sqrt{f_v^2(i,j)+f_h^2(i,j)} \quad \text{.... (1)}$$

$$\Theta=\tan^{-1}\left(f_v(i,j)/f_h(i,j)\right) \quad \text{..... (2)}$$

-1	0	1
-2	0	2
-1	0	1

1	2	1
0	0	0
-1	-2	-1

Fig.10.(a)Horizontal Sobel operator (b) Vertical Sobel operator

C. Wavelet Feature

Wavelet feature has some beneficial properties which make it very robust. It compacts most of the signal's energy into its wavelet coefficients, it effectively represent low frequency components (such as image background) as well as high frequency transients (such as image edges) and it has ability of a progressive transmission, which facilitates the reception of an image at different qualities. The coefficients give scale invariant representation in multi resolution analysis, as all the important characteristics of the image are being incorporated in the wavelet coefficients. Here Discrete wavelet transform (DWT) was used in which 2nd level 2d Haar wavelet was used to get the decomposition vector which outputs horizontal, vertical and diagonal decomposition vectors. The Haar wavelet's mother wavelet function and scaling function are given below and are taken from Wikipedia.

$$\psi(t) = \begin{cases} 1 & 0 \le t < \frac{1}{2} \\ -1 & \frac{1}{2} \le t < 1 \\ 0 & otherwise \end{cases} \qquad \phi(t) = \begin{cases} 1 & 0 \le t < 1 \\ 0 & otherwise \end{cases} \quad \ldots (3)$$

D. Normalizing feature Vector:

Each of these feature obtained have values in some random range which makes the classification very Non Uniform. So before making a final feature vector they were normalized to a confined range of values. Range was taken from [-1, 1].

E. Final Feature Vector

All the above three features were concatenated for every letter to obtain the final feature vector.

Final vector = [[Distance Feature].[Gradient Feature].[Wavelet Feature]]

This feature vector was calculated for all the characters present in the database, and each character type was appended with a label at the beginning of the feature vector, and all the values are stored in a text file which acts as an input to train the SVM machines.

IX. Learning Theory

A. SVM with kernel

SVM are based on the Structural Risk Minimization principle which means the hypothesis obtained will have the lowest true error. Here, kernel method was used with kernel function as Radial basis function. Below equations are taken from stand

B. Equations

$$\frac{\partial}{\partial w} L_P = w_v - \sum_i \alpha_i y_i w_{iv} \qquad v = 1, \ldots, d \quad \ldots (4)$$

$$\frac{\partial}{\partial b} L_P = -\sum_i \alpha_i y_i \qquad \alpha_i y_i = 0 \quad \ldots (5)$$

$$y_i \left(\vec{x_i} . \vec{w} + b \right) - 1 \ge 0 \qquad i = 1, \ldots, l \quad \ldots (6)$$

$$\alpha_i \ge 0 \quad for \quad all \quad i$$

$$I\alpha_i \left(y_i \left(\vec{x_i} . \vec{w} + b \right) - 1 \right) = 0 \quad for \quad all \quad i \quad \ldots (7)$$

Radial Basis Kernel Function:

$$K(x, x') = \exp\left(- \| x - x' \|^2 / 2\sigma^2 \right) \quad \ldots (8)$$

$\|x-x'\|_2$ may be recognized as the squared Euclidean distance between the two feature vectors. σ =1 is a free parameter and $\gamma = -1/2\sigma^2$

$$K(x, x') = \exp\left(\gamma \| x - x' \|^2 \right) \quad \ldots (9)$$

C. LIBSVM

Here LIBSVM (Library for Support Vector Machine) was used. It supports multiclass classification. Some of the functions of the library are discussed below.

Libsvmwrite ('Name.train', labels, sparse_matrix)

Using this the training data had been converted to LIBSVM format under the name Name.train. Next, the obtained Name.train was read using libsvmread ('Name.train') to obtain the label matrix and the feature matrix separately. These matrices were then passed to svmtrain () function to train the SVM classifier. Below the svmtrain function is briefed

Svmtrain (label_matrix, feature_matrix,'-c 1 -g 0.07 -t 2')

Attributes of svm in this function define various characteristics of the SVM, some are –s define svm-type,-t is kernel-type, -d is degree,-g is gamma in kernel function, -c cost etc. these are some of the parameters whose appropriate values defines the SVM model characteristics.

D. SVM Models

Above explained strategy for SVM was applied to three separate SVM Models. They all were customized for three different obtained regions of a character namely upper region, middle region and lower region.

First, for the upper region, this model classified all the upper modifiers or 'Matras' together.

Second, for the middle region, this model classified all the consonants and vowels of the script and it also helped in distinguishing some type modifiers such as 'ee', 'aa' and 'eee'. Also it helped in classification of half letters.

Third, for the lower region, this model classified all the lower modifiers or 'Matras' together.

All these models were trained with the handwritten database of 10 to 15 different persons; about 50 to 70 characters of each type.

X. Text to Speech System

A. Digitized text to speech

The labels coming from SVM were first digitized using Unicode. The Unicode representations were stored in an array. Three trees were formed to receive the codes from the array to get concatenated.

B. Labels to Unicode

Since digital understanding is ingrained in Unicode, ASCII or other conventions, only those symbols that are represented by these standards were classified; all of them covering a typical writing style. The recognized symbols (or parts of syllables) classified by svm were combined to produce corresponding syllables. These symbols were grouped according to the order of recognition. As symbol labels came sequentially from SVM, their corresponding Unicode representations were stored in a file.

C. Speech file synthesis

Three independent trees were implemented for three stripes. Unicode data from file were accessed sequentially and given to the trees in their order of occurrence to append speech file by associated phoneme file. Index on the file contents was incremented by one, each tree was called one after another. In increments of three, one tree received an input Contents of the file was then printed on computer screen

XI. CONCLUSION AND RESULTS

A. Training

Each character in the training is a 45*45 pixel image. All the above processes of feature extraction are applied to this image to obtain array of 870 features per character. The Training was done on two different processor to obtain following timings.

Model	No. of Characters	No. of character of Each type	Total Characters	Time taken for the training
Upper	5	15	75	2.29284 s
Middle	74	50	3700	93.0.291s
Lower	2	15	30	1.292971 s

Above mentioned training was done on i5 processor. While in case of i7 the total timing is reduced to 66.25s which further reduces in case of GPU.

B. Testing

Here the testing was done in 2 ways. First by passing random characters to the OCR system, the recognition rate was found to be almost 100 %. All the characters were efficiently recognized. But secondly, a whole document was scanned and passed to OCR then the rate of recognition slightly decreases to 96 %. This happens due mainly overlapping of characters.

C. Speech synthesis Results

The efficiency of TTS system directly depends upon the efficiency of OCR i.e. all the letters correctly recognized by OCR gets the correct speech signal associated with it. Thus concatenation of all such signals together make a wave file for the whole document which can be saved for later purposes. Here the voice produced was without prosodic effects.

X. REFERENCES

[1] http://tdil.mit.gov.in

[2] Badhika, Sirisha. "Implementation of Multilevel Segmentation using Cognitive Approach-A Case study on Devanagari Script." International Journal of Scientific and Research Publications: 335.

[3] Gupta, Shaina, and Daulat Sihag. "RECOGNITION OF HANDWRITTEN DEVNAGARI NUMERALS WITH SVM CLASSIFIER." (2014).

[4] K. Elissa, “Title of paper if known,” unpublished.

[5] Pal, U., and B. B. Chaudhuri. "Indian script character recognition: a survey." *Pattern Recognition* 37, no. 9 (2004): 1887-1899..

[6] Zhang, Weipeng, Yuan Yan Tang, and Yun Xue. "Handwritten character recognition using combined gradient and wavelet feature." In *Computational Intelligence and Security, 2006 International Conference on*, vol. 1, pp. 662-667. IEEE, 2006.

[7] Rafael C. Gonzalez, Richard E. Wood Book. Digital Image Processing 2nd edition pp. 349-404.

[8] Hsu, Chih-Wei, and Chih-Jen Lin. "A comparison of methods for multiclass support vector machines." *Neural Networks, IEEE Transactions on* 13, no. 2 (2002): 415-425.

[9] Burges, Christopher JC. "A tutorial on support vector machines for pattern recognition." *Data mining and knowledge discovery* 2, no. 2 (1998):121-167.

[10] http://www.csie.ntu.edu.tw/~cjlin/libsvm/faq.html.

[11] Shimizu, A., K. Hachimine, N. Ohki, H. Ohta, M. Koguchi, Y. Nonaka, H. Sato, and F. Ootsuka. "Local mechanical-stress control (LMC): A new technique for CMOS-performance enhancement." In *Electron Devices Meeting, 2001. IEDM'01. Technical Digest. International*, pp. 19-4. IEEE, 2001.

[12] Sarathy, Konakanchi Partha, and A. G. Ramakrishnan. "A research bed for unit selection based text to speech synthesis." In *Spoken Language Technology Workshop, 2008. SLT 2008. IEEE*, pp. 229-232. IEEE, 2008.