A Texture Based Methodology for Text Region Extraction from Low Resolution Natural Scene Images

S. A. Angadi vinay_angadi@yahoo.com
Department of Computer Science & Engineering
Basaveshwar Engineering College
Bagalkot, 587102, Karnataka, India

M. M. Kodabagi malik123_mk@rediffmail.com
Department of Computer Science & Engineering
Basaveshwar Engineering College
Bagalkot, 587102, Karnataka, India

Abstract

Automated systems for understanding display boards are finding many applications useful in guiding tourists, assisting visually challenged and also in providing location aware information. Such systems require an automated method to detect and extract text prior to further image analysis. In this paper, a methodology to detect and extract text regions from low resolution natural scene images is presented. The proposed work is texture based and uses DCT based high pass filter to remove constant background. The texture features are then obtained on every 50x50 block of the processed image and potential text blocks are identified using newly defined discriminant functions. Further, the detected text blocks are merged and refined to extract text regions. The proposed method is robust and achieves a detection rate of 96.6% on a variety of 100 low resolution natural scene images each of size 240x320.

Keywords: Text Region Extraction, Texture Features, Low Resolution Natural scene image.

1. INTRODUCTION

As the people move across world for business, field works and/or pleasure, they find it difficult to understand the text written on display boards in foreign environment. In such a scenario, people either look for guides or intelligent devices that can help them in providing translated information to their native language. As most of the individuals carry camera embedded, hand held devices such as mobile phones and PDA's, there is a possibility to integrate technological solutions into such systems inorder to provide facilities for automatically understanding display boards in foreign environment. These facilities may be provided as an integral solution through web service as necessary computing function, which are not available in hand held systems. Such web based hand held systems must be enabled to capture natural scene images containing display boards and query the web service to retrieve translated localized information of the text written on display boards.

The written matter on display/name boards provides information necessary for the needs and safety of people, and may be written in languages unknown. And the written matter can be street names, restaurant names, building names, company names, traffic directions, warning signs etc. Hence, lot of commercial and academic interest is veered towards development of techniques for web service based hand held systems useful in understanding written text in display boards. There is a spurt of activity in development of web based intelligent hand held tour guide systems, blind assistants to read written text and Location aware computing systems and many more in recent years. A few such works are presented in the following and a more elaborate survey of related works is given in the next section. *A point by photograph paradigm* where users can specify an object by simply taking picture to retrieve matching images from the web is found in [1]. The *comMotion* is a location aware hand held system that links personal information to locations. It reminds users about shopping list when he/she nears a shopping mall [2]. At Hewlett Packard (HP), mobile Optical Character Reading (OCR) applications were developed to retrieve information related to the text image captured through a pen-size camera [3]. Mobile phone image matching and retrieval has been used by insurance and trading firms for remote item appraisal and verification with a central database [4].

The image matching and retrieval applications cannot be embedded in hand held devices such as mobile phones due to limited availability of computing resources, hence such services are being developed as web services. The researchers have also worked towards development of web based intelligent hand held tour guide systems. The cyberGuide [5] is an intelligent hand held tour guide system, which provides the information based on user's location. The cyberGuide continuously monitors the users location using Global Positioning System (GPS) and provides new information at the right time. Museums could provide these tour guides to visitors allowing them to take personalized tours observing any displayed object. As the visitors move across museum floors, the information about the location is pushed to hand held tour guides. The research prototypes used to search information about an object image captured by cameras embedded in mobile phones are described in [6-7].

The state of art hand held systems available across the world are not automated for understanding written text on display boards in foreign environment. Scope exists for exploring such possibilities through automation of hand held systems. One of the very important processing steps for development of such systems is automatic detection and extraction of text regions from low resolution natural scene images prior to further analysis. The written text provides important information and it is not an easy problem to reliably detect and localize text embedded in natural scene images [8]. The size of the characters can vary from very small to very big. The font of the text can be different. Text present in the image may have multiple colors. The text may appear in different orientation. Text can occur in a complex background. And also the textual and other information captured is affected by significant degradations such as perspective distortion, blur, shadow and uneven lighting. Hence, the automatic detection and segmentation of text is a difficult and challenging problem. Reported works have identified a number of approaches for text localization from natural scene images. The existing approaches are categorized as connected component based, edge based and texture based methods. Connected component based methods use bottom up approach to group smaller components into larger components until all regions are identified in the image. A geometrical analysis is later needed to identify text components and group them to localize text regions. Edge based methods focus on the high contrast between the background and text and the edges of the text boundary are identified and merged. Later several heuristics are required to filter out nontext regions. But, the presence of noise, complex background, and significant degradation in the low resolution natural scene image can affect the extraction of connected components and identification of boundary lines, thus making both the approaches inefficient. Texture analysis techniques are good choice for solving such a problem as they give global measure of properties of a region.

In this paper, a new texture based text detection and segmentation method is proposed. The proposed method uses high pass filtering in the DCT domain to suppress most of the background. Later texture features [33] such as *homogeneity* and c*ontrast* are computed on image blocks to identify and segment text regions in the image. Each unit block is classified as either text or nontext based on newly defined discriminant functions. In addition, merging algorithms are used to merge text blocks to obtain text regions. The regions are further refined using post processing. The proposed method is robust enough to detect text regions from low resolution natural scene images, and achieves a detection rate of 96.6%. The system is developed in MATLAB and evaluated for 100 low resolution natural scene images on Intel Celeron (1.4GHz) computer. It was observed that the processing time lies in the range of 6 to 10 seconds due to varying background. The proposed methodology is described in the following sections of the paper.

The rest of the paper is organized as follows; the detailed survey related to text extraction from natural scene images is described in section 2. The proposed method is presented in Section 3. The experimental results and analysis are given in Section 4. Section 5 concludes the work and lists future directions.

2. RELATED WORK

The web based hand held systems useful in understanding display boards requires analysis of natural scene images to extract text regions for further processing. A number of methods for text localization have been published in recent years and are categorized into connected component based, edge based and texture based methods. The performance of the methods belonging to first two categories is found to be inefficient and computationally expensive for low resolution natural scene images due to the presence of noise, complex background and significant degradation. Hence, the techniques based on texture analysis have become a good choice for image analysis, and texture analysis is further investigated in the proposed work.

A few state of the art approaches that use texture features for text localization have been summarized here; the use of horizontal window of size 1×21 (Mask size) to compute the spatial variance for identification of edges in an image, which are further used to locate the boundaries of a text line is proposed in [9]. However, the approach will only detect horizontal components with a large variation compared to the background and a processing time of 6.6 seconds with 256x256 images on SPARC station 20 is reported. The Vehicle license plate localization method that uses similar criteria is presented in [10]. It uses time delay neural networks (TDNNs) as a texture discriminator in the HSI color space to decide whether the window of an image contains a license plate number. The detected windows are later merged for extracting license plates. A multi-scale texture segmentation schemes are presented in [11-12]. The methods detect potential text regions based on nine second-order Gaussian derivatives and is evaluated for different images including video frames, newspapers, magazines, envelopes etc. The approach is insensitive to the image resolution and tends to miss very small text and gives a localization rate of 90%.

A methodology that uses frequency features such as the number of edge pixels in horizontal and vertical directions and Fourier spectrum to detect text regions in real scene images is discussed in [13]. The texture-based text localization method using Wavelet transform is proposed in [14]. The techniques for text extraction in complex color images, where a neural network is employed to train a set of texture discrimination masks that minimize the classification error for the two texture classes: text regions and non-text regions are reported in [16-17].

Learning-based methods for localizing text in documents and video are proposed in [15] and [18]. The method [18] is evaluated for various video images and the text localization procedure required about 1 second to process a 352x240 image on Sun workstation. And for text detection a precision rate of 91% and a recall rate of 92.8% is reported. This method was subsequently enhanced for skewed text in [19]. A work similar to the proposed method which uses DCT coefficients to capture textural properties for caption localization is presented in [8]. The authors claim that the method is very fast and gives a detection rate of 99.17%. However, the precise localization results are not reported. In recent years, several approaches for sign detection, text detection and segmentation from natural images are also reported [20-32].

Out of many works cited in the literature it is generally agreed that the robustness of texture based methods depends on texture features extracted from the window/block or the region of interest that are used by the discriminant functions for classification decisions. The probability of misclassification is directly related to the number and quality of details available in texture features. Hence, the extracted texture features must give sufficient information to distinguish text from the background. And also suppression/removal of background information is an essential preprocessing step needed before extracting distinguishable texture features to reduce the probability of misclassification. But most of the works cited in the literature directly operate on the image without suppressing the background. Hence, there is a scope to explore such possibilities. The proposed method performs preprocessing on the image for suppressing the uniform background in the DCT domain and further uses texture features for text localization. The detailed description of the proposed methodology is given in the next section.

3. TEXTURE BASED METHODOLOGY FOR TEXT EXTRACTION

The proposed methodology is texture based, and operates on low resolution natural scene images captured by cameras embedded in mobile phones to detect and segment text regions. The methodology uses high pass filter in the DCT domain to suppress the background, and texture features such as *homogeneity* and *contrast* to detect and segment text regions. The processing is carried out on 8x8 sized image blocks during background suppression phase and the remaining phases use 50x50 sized image blocks. There are several benefits of using larger sized image blocks for extracting texture features. One such benefit is, the larger size image blocks cover more details and hence extracted features give sufficient information for correct classification of blocks into text and nontext categories. The other benefits include; robustness and insensitiveness to variation in size, font and alignment.

The proposed method comprises of 5 phases; Background removal/suppression in the DCT domain, texture features computation on every 50x50 block and obtaining a feature matrix *D*, Classification of blocks, merging of text blocks to detect text regions, and refinement of text regions. The block schematic diagram of the proposed model is given in figure 1. The detailed description of each phase is presented in the following subsections;

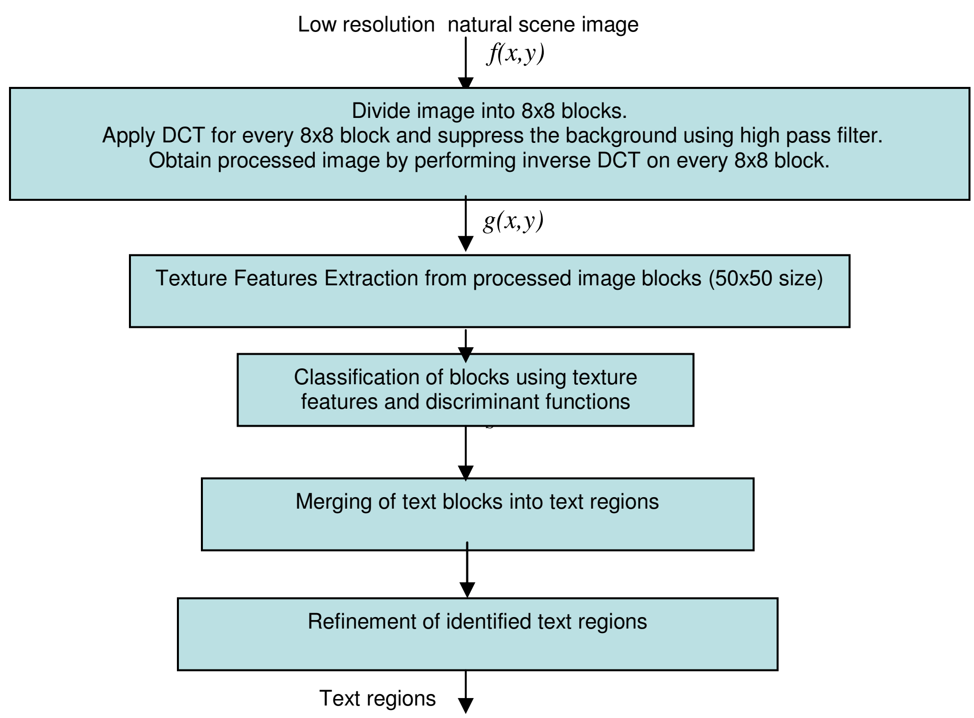

FIGURE 1: Block diagram of proposed method

3.1 Background removal/suppression in the DCT domain

The removal of constant background from the low resolution natural scene image resulting from sources such as building walls, windows, trees, and others is an essential preprocessing step, required to reduce the effort needed for further image analysis. The proposed method uses DCT coefficients to remove constant background. The DCT coefficient values are computed on every 8x8 block of the image. In the corresponding DCT block, the values from top to bottom indicate horizontal variations, with increasing frequencies. The value at the top left corner (first entry in the DCT matrix/block) corresponds to DC component. And the values from left to right indicate vertical variations, with increasing frequencies. Therefore the top left values which represent low frequency components contain most of the energy, while the high frequency components that are located towards the bottom right corner are mostly blank (contain zero values). Hence, the constant background is removed successfully by applying a high pass filter that attenuates the DC component of every 8x8 DCT block of the image *f(x ,y)* of size *L x M*, where *x* and *y* are spatial coordinates. The transform function of high pass filter that operates on every 8x8 DCT block is given in equation 1. Later the processed/background suppressed image *g(x,y)* is obtained by applying inverse DCT on every 8x8 DCT block, which will be used in subsequent phases. The steps of background suppression are depicted in equations 2, 3 and 4. The block diagram of background suppression using DCT is given in figure 2.

$$H(u,v) = \begin{cases} 0 & (u,v)=(1,1), \text{ where } u=1\ldots8,\ v=1\ldots8 \\ 1 & \text{Otherwise} \end{cases} \quad (1)$$

The high pass filter attenuates the DC component by storing value zero in every top left coordinate of 8x8 DCT block. This process is also called removing low frequency component from top left corner from every DCT block.

$$G(u,v) = DCT[f(x,y)] \quad \text{where } 1 >= x,u <= L, \text{ and } 1 >= y,v <= M \quad (2)$$
$$P(u,v) = H(u,v)\, G(u,v). \quad (3)$$
$$g(x,y) = DCT^{-1}[P(u,v)]. \quad (4)$$

Where,
G(u,v) is DCT matrix of input image *f(x,y)*.
P(u,v) is Processed DCT matrix.
g(x,y) is background suppressed image.

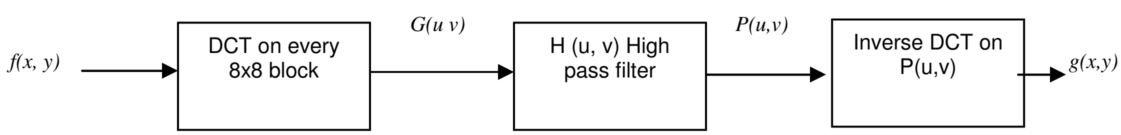

FIGURE 2: High Pass Filter for Background Removal using DCT

3.2 Features Extraction

In this phase, the *texture* features such as *homogeneity* and *contrast* are obtained from every 50 x 50 block of the processed image *g(x,y)* at $0^0, 45^0, 90^0$, and 135^0 orientations. Totally 8 features are extracted from every block and are stored into a feature vector X_i (Subscript "i" corresponds to i^{th} block). The feature vector X_i also records block coordinates which corresponds to minimum and maximum row and column numbers of the block. Feature vectors of all *N* blocks are combined to form a feature matrix *D* as depicted in equation 5. The feature vector is described in equation 6.

$$D = [X_1, X_2, X_3 \ldots\ldots X_N]^T \quad (5)$$
$$X_i = [r_{min}, r_{max}, c_{min}, c_{max}, f_j, j=1,8]; \quad (6)$$

Where;

r_{min}, r_{max}, c_{min}, c_{max} corresponds to coordinates of i^{th} block in terms of minimum and maximum row and column numbers.
f_1 and f_2 corresponds to homogeneity and contrast at 0 degree orientation.
f_3 and f_4 corresponds to homogeneity and contrast at 45 degree orientation.
f_5 and f_6 corresponds to homogeneity and contrast at 90 degree orientation.
f_7 and f_8 corresponds to homogeneity and contrast at 135 degree orientation.

The features *homogeneity* and *contrast* are calculated as in equations 7 and 8.

$$Homogeneity = \sum_{i=1}^{Q} \sum_{j=1}^{Q} (P(i,j)/R)^2 \quad (7)$$

$$Contrast = \sum_{n=1}^{Q-1} n2 \sum_{|i-j|=n}^{Q} (P(i,j)/R) \quad (8)$$

Where R is given in equation 9.

$$R = \sum_{i=1}^{Q} \sum_{j=1}^{Q} P(i,j) \quad (9)$$

P corresponds to cooccurence matrix at a given degree.
R is normalized value of cooccurrence matrix P.
N is total number of blocks.
QxQ is dimension of block size which is chosen as 50x50.

3.3 Classification

The classification phase of the proposed model uses discrimanant functions to classify every block into two classes' *w1* and *w2* based on feature vector *Xi*. Where, *w1* corresponds to text blocks and *w2* corresponds to nontext blocks category. The discriminant functions uses two thresholds *T1* initialized to 0.4 and *T2* to 50 corresponding to *homogeneity* and *contrast* values respectively. The values 0.4 for *T1* and 50 for *T2* are heuristics chosen based on experiments conducted on several different images and are used by classifiers to produce correct classification results. The discriminant functions *d1* and *d2* together decides a block as text block if the *homogeneity* and *contrast* features values at 0^0,45^0,90^0, and 135^0 orientations are less than thresholds *T1* and *T2*. The classification rules using discriminant functions are stated in equations 10 and 11.

Given a feature matrix D of features Xi, assign the corresponding image block to;

Class *w1* if *d1(X_i) is satisfied, and d2(X_i) is satisfied*
Class *w2* *otherwise*
Where,
d1(X_i) is a discriminant function which defines/specifies constraint on homogeneity value.
d2(X_i) is a discriminant function which defines/specifies constraint on Contrast value.

$$d1(X_i) = \begin{cases} \text{satisfied} & \text{if } X_i\,(fj) <= T1,\ \forall\ i=1, N \text{ and } j = 1,3,5,7 \\ \text{Not satisfied} & \text{otherwise} \end{cases} \quad (10)$$

$$d2(X_i) = \begin{cases} \text{satisfied} & \text{if } X_i(fj) >= T2\ \ \forall\ i=1,N \text{ and } ,j = 2,4,6,8 \\ \text{Not satisfied} & \text{otherwise} \end{cases} \quad (11)$$

Where,
T1 corresponds to threshold on homogeneity (T1 = 0.4), chosen empirically.
T2 corresponds to threshold on contrast T2 = (50), chosen empirically.

The coordinates r_{min}, r_{max}, c_{min}, c_{max} which corresponds to minimum and maximum row and column numbers of every classified text block C_i will be stored into a new vector B as in equation 12, which are later used during merging process to obtain contiguous text regions.

$$B = [C_i,\ i = 1, N1] \tag{12}$$

$$C_i = [r_{min}, r_{max}, c_{min}, c_{max}] \tag{13}$$

Where,

C_i *corresponds to i^{th} text block.*

$r_{min}, r_{max}, c_{min}, c_{max}$ *corresponds to the coordinates of i^{th} block in terms of minimum and maximum row and column numbers.*

$N1$ *is Number of text blocks.*

The thresholds *T1* and *T2* are highly dependent on the size of the block. For 50 X 50 block size the values 0.4 for *T1* and 50 for *T2* have given correct classification results and have been chosen after exhaustive experimentation. It is also found during the experiments that, the probability of misclassification decreases by careful selection of values for *T1* and *T2*. After this phase, the classified text blocks are subjected to merging process as given in the next section.

3.4 Merging of text blocks to detect text regions

The merging process combines the potential text blocks C_i, connected in rows and columns, to obtain new text regions r_i, whose coordinates are recorded into vector R as depicted in equation 14.

$$R = [r_i,\ i = 1, W] \tag{14}$$

$$r_i = [r_{min}, r_{max}, c_{min}, c_{max}] \tag{15}$$

r_i *corresponds to i^{th} text region*

$r_{min}, r_{max}, c_{min}, c_{max}$ *corresponds to the coordinates of i^{th} text region.*

W *is number of text regions.*

The merging procedure is described in algorithm 1;

Algorithm1

Input**:* Vector ***B which contains coordinates of identified text blocks

Output**:* Vector ***R which records text regions

Begin

1. Choose the first block $\boldsymbol{C_s}$ from vector ***B***.
2. Initialize coordinates of a new text region $\boldsymbol{r_i}$ to coordinates of ***block*** $\boldsymbol{C_s}$.
3. **Se**lect next block $\boldsymbol{C_p}$ from the vector ***B***.
4. ***if*** (the block C_p is connected to $\boldsymbol{r_i}$ *in* row or column) ***then***

   begin

   Merge and update coordinates r_{min}, r_{max}, c_{min}, c_{max} of block $\boldsymbol{r_i}$.

   $r_{min} = \min\{\, \mathbf{ri}\, [r_{min}],\ C_p[r_{min}] \,\}$ $\quad r_{max} = \max\{\, \mathbf{ri}\, [r_{max}],\ C_p[r_{max}] \,\}$

   $c_{min} = \min\{\, \mathbf{ri}\, [c_{min}],\ C_p[c_{min}] \,\}$ $\quad c_{max} = \max\{\, \mathbf{ri}\, [c_{max}],\ C_p[c_{max}] \,\}$

   else

   Store text region $\boldsymbol{r_i}$ into vector ***R***.

   Initialize coordinates of a new text region $\boldsymbol{r_i}$ to coordinates of current *block* $\boldsymbol{C_p}$.

   end
5. Repeat steps *2 -5* until *p=N1.*

End

3.5 Refinement of text regions

The refinement phase is a post processing step used to improve the detection accuracy. This phase is concerned with refining the size of the detected text regions to cover small portions of missed text present in adjacent undetected blocks and unprocessed regions. The refinement process is carried out in two steps; Adjacent undetected blocks processing and Combining unprocessed region. The functionality of each step is described below;

3.5.1 Adjacent undetected blocks processing

In this step, every detected text region r_i is refined either by combining the entire size (50 rows and 50 columns) or selected rows and columns of adjacent undetected blocks (in rows and columns) which contain/cover small portions of missed text. The *average contrast* value (f2+f4+f6+f8) / 4) is computed for every adjacent undetected block (in row or column), If the computed value is greater than or equal to *35* then the entire block size is combined with region r_i for refinement assuming that the missed adjacent block may contain significant text information. The heuristic value *35* is chosen empirically based on experiments. Similarly, if the *average contrast* value is between 5-9, 10-19, and 20-34, then *5, 10, and 20* respective adjacent rows/columns are added to region r_i for refinement. Again the heuristic values *5, 10 and 20* are chosen empirically based on experiments and results were encouraging. The procedure for combining adjacent undetected blocks is described in algorithm 2.

***Algorithm** 2*

Input**:* - Vector ***R which contains coordinates of extracted text regions.
- Vector ***D*** which contains texture features and coordinates of all 24 blocks of preprocessed image.
- Unprocessed 40 rows and 20 columns of the preprocessed image

Output**:* Vector ***R which records refined text regions.

Begin

1. Start with first text region r_i.

2. Select the next feature vector X_p from the feature matrix ***D***.

3. ***if*** (the feature vector ***Xp*** *is* connected to text region r_i *in row or column)* ***then***

begin

3.1 Find average *contrast* value = (f2+f4+f6+f8) / 4.

3.2 ***if*** (average *contrast* value is > 35) ***then***

begin

Merge and update coordinates $r_{min}, r_{max}, c_{min}, c_{max}$ of text region r_i by adding adjacent block.

$r_{min} = min\{ ri[r_{min}], C_p[r_{min}] \}$ $r_{max} = max\{ ri[r_{max}], C_p[r_{max}] \}$

$c_{min} = min\{ ri[c_{min}], C_p[c_{min}] \}$ $c_{max} = max\{ ri[c_{max}], C_p[c_{max}] \}$

else

if (average *contrast* value is between 5 to 9) ***then***

begin

$c_{min} = c_{min} - 5$; *// add 5 columns if* feature vector ***Xp*** *left connected*

or

$c_{max} = c_{max} + 5$; *// add 5 columns if* feature vector ***Xp*** *right connected*

or

$r_{min} = r_{min} - 5$; *// add 5 rows if* feature vector ***Xp*** *top connected*

or

$r_{max} = r_{max} + 5$; *// add 5 rows if* feature vector ***Xp*** *bottom connected*

end

if (average *contrast* value is between 10 to 19) ***then***

begin

$c_{min} = c_{min} - 10$; *// add 5 columns if* feature vector ***Xp*** *left connected*

or

$c_{max} = c_{max} + 10$; *// add 5 columns if* feature vector ***Xp*** *right connected*

or

$r_{min} = r_{min} - 10$; *// add 5 rows if* feature vector ***Xp*** *top connected*

or

$r_{max} = r_{max} + 10$; *// add 5 rows if* feature vector ***Xp*** *bottom connected*

end

if (average *contrast* value is between 20 to 34) ***then***

begin

$c_{min} = c_{min} - 20$; *// add 5 columns if* feature vector ***Xp*** *left connected*

or

$c_{max} = c_{max} + 20$; *// add 5 columns if* feature vector ***Xp*** *right connected*

or

$r_{min} = r_{min} - 20$; *// add 5 rows if* feature vector ***Xp*** *top connected*

or

$r_{max} = r_{max} + 20$; *// add 5 rows if* feature vector ***Xp*** *bottom connected*

end

end

4. Repeat steps *2 -3 until p = N1.*

5. Select next text region $r_i = [r_{min}, r_{max}, c_{min}, c_{max}]$ *from* ***R***.
6. Repeat steps 2-5 until all text regions are refined.

End

3.5.2 Combining unprocessed regions

The proposed method works by dividing an image of size 240x320 into 50x50 sized blocks till the completion of phase 3.3. Hence the remaining 40 rows and 20 columns will be left unprocessed after phase 3.3. As the unprocessed rows and columns may also contain text information they need to be processed for further refinement of detected text regions. In this step, the detected text regions are further refined by processing and adding adjacent unprocessed area to cover missed portion containing small text. During processing only *average contrast* feature value (f2+f4+f6+f8) / 4) is computed from unprocessed area, and if the *average contrast* value is greater than or equal to the threshold *50,* then the entire size of the unprocessed region is combining with adjacent detected text regions for refinement. The heuristic value *50* is chosen empirically based on experiments and has produced good results.

The proposed methodology is robust and performs well for different sizes of font and image resolution. The block size is an important design parameter whose dimension must be properly chosen to make the method more robust and insensitive to variation in size, font and its alignment. The approach also detects nonlinear text regions and results are presented in the next section. However, the method detects larger text regions than the actual size when the image background is more complex containing trees, vehicles, and other details from sources of outdoor scenes. The method takes about 6 to 10 seconds of processing time based on the complexity of background contained in the image.

4. RESULTS AND ANALYSIS

The proposed methodology for text region detection and extraction has been evaluated for 100 indoor and outdoor low resolution natural scene display board images (having 2400, 50x50 size blocks) with complex backgrounds. The experimental tests were conducted for most of the images containing Kannada text and few containing English text and results were highly encouraging. The experimental results of processing a typical display board image containing text with varying background is described in section 4.1. And the results of processing several other display board images dealing with various issues and the overall performance of the system are reported in section 4.2.

4.1 Text Region Extraction of a typical display board image

A display board image of size 240x320 given in figure 3a, containing text information having smaller and bigger Kannada characters and complex backgrounds such as building walls, doors, and uneven lighting conditions is initially preprocessed to suppress the background using a high pass filter in the DCT domain. The experimental values of applying a high pass filter for the first 8x8 DCT block of image in figure 3a are given in Table 1.

FIGURE 3: a) Original image b) Background Suppressed image c) Detected text region before post processing d) Refined text region after post processing

TABLE1: Preprocessing of first 8x8 block of an image in figure 3a

Step1: 8x8 first image block	**Step2: 8x8 first DCT block**
159 159 159 159 159 159 159 159	1.2821 -0.0019 0.0007 -0.0006 0.0004 -0.0002 0.0001 0.0002
159 159 159 159 159 159 159 159	-0.0062 0.0009 -0.0001 0.0001 0.0001 -0.0003 0.0002 -0.0002
160 160 160 160 160 160 160 161	-0.0015 0.0013 -0.0004 0.0003 -0.0004 0.0004 -0.0003 -0.0000
160 160 160 160 160 160 160 161	-0.0001 -0.0011 -0.0002 -0.0000 -0.0002 0.0003 -0.0001 0.0003
160 160 160 160 161 161 161 162	-0.0006 0.0003 -0.0001 0.0003 0.0001 -0.0004 0.0002 -0.0001
160 160 161 161 161 161 162 162	0.0007 0.0002 0.0004 -0.0003 0.0000 0.0002 -0.0001 -0.0001
161 161 161 161 161 161 162 162	0.0007 -0.0002 -0.0003 -0.0003 0.0000 0.0001 0.0001 0.0000
161 161 161 161 161 161 161 161	0.0009 -0.0003 0.0004 0.0002 0.0001 -0.0004 0.0000 -0.0000
Step 4: 8x8 first preprocessed image block after applying inverse DCT	**Step3: 8x8 first DCT block after applying high pass filter**
0 0 0 0 0 0 0 0	0 -0.0019 0.0007 -0.0006 0.0004 -0.0002 0.0001 0.0002
0 0 0 0 0 0 0 0	-0.0062 0.0009 -0.0001 0.0001 0.0001 -0.0003 0.0002 -0.0002
0 0 0 0 0 0 0 1	-0.0015 0.0013 -0.0004 0.0003 -0.0004 0.0004 -0.0003 -0.0000
0 0 0 0 0 0 0 1	-0.0001 -0.0011 -0.0002 -0.0000 -0.0002 0.0003 -0.0001 0.0003
0 0 0 0 1 1 1 2	-0.0006 0.0003 -0.0001 0.0003 0.0001 -0.0004 0.0002 -0.0001
0 0 1 1 1 1 2 2	0.0007 0.0002 0.0004 -0.0003 0.0000 0.0002 -0.0001 -0.0001
1 1 1 1 1 1 2 2	0.0007 -0.0002 -0.0003 -0.0003 0.0000 0.0001 0.0001 0.0000
1 1 1 1 1 1 1 1	0.0009 -0.0003 0.0004 0.0002 0.0001 -0.0004 0.0000 -0.0000

The experimental values in Table 1 demonstrate that the constant gray level values in the range 159-162 in the first 8x8 block of the image in figure 3a have been compressed to a narrow range 0-2 after preprocessing. Hence, the transform function given in equation 1 that attenuates DC component as given in step3 has performed well in suppressing most of the constant background. The processed image after applying background suppression for all blocks is shown in figure 3b, where most of the unwanted details are removed. And only gray level discontinuities belonging to text and edges remain for further image analysis. Hence, the extracted texture features such as *homogeneity* and *contrast* from such preprocessed images aid classification decisions and help in increasing detection rate. All 24 50x50 image blocks of preprocessed image in figure 3b from which texture features are extracted are shown in figure 3e. Table 2 summarizes the extracted features.

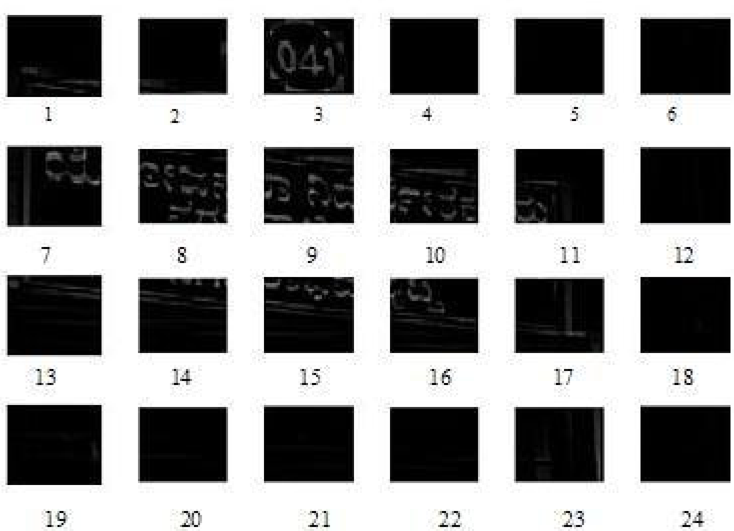

FIGURE 3e: 24 50x50 blocks of preprocessed image in figure 3b.

TABLE2: Feature Matrix D containing extracted features of image of size 240x320 given in figure 3b.

block	r_{min}	r_{max}	c_{min}	c_{max}	***f1***	***f2***	***f3***	***f4***	***f5***	***f6***	***f7***	***f8***
1	1	50	1	50	0.35563	5.7518	0.27407	40.044	0.29799	33.25	0.27741	33.556
2	1	50	51	100	0.37939	3.9122	0.30362	25.075	0.32683	21.255	0.31086	22.454
3	***1***	***50***	***101***	***150***	***0.23571***	***145.05***	***0.21203***	***256.51***	***0.26159***	***115.69***	***0.21793***	***224.92***
4	1	50	151	200	0.5486	0.060408	0.4624	0.14452	0.49457	0.11714	0.46736	0.13869
5	1	50	201	250	0.55128	0.056735	0.47864	0.11308	0.51388	0.081837	0.48201	0.11058
6	1	50	251	300	0.41841	0.21367	0.35536	0.33944	0.40281	0.19714	0.35614	0.33882
7	***51***	***100***	***1***	***50***	***0.23662***	***80.981***	***0.20995***	***134.79***	***0.2649***	***59.38***	***0.21274***	***121.14***
8	***51***	***100***	***51***	***100***	***0.29437***	***108.49***	***0.23146***	***266.22***	***0.27296***	***167.09***	***0.24404***	***213.76***
9	***51***	***100***	***101***	***150***	***0.27832***	***94.329***	***0.21101***	***271.21***	***0.24637***	***182.34***	***0.23078***	***220.13***
10	***51***	***100***	***151***	***200***	***0.26405***	***70.58***	***0.1898***	***185.71***	***0.22773***	***130.81***	***0.20226***	***169.87***
11	***51***	***100***	***201***	***250***	***0.35865***	***31.801***	***0.29939***	***78.564***	***0.35312***	***47.617***	***0.31177***	***60.969***
12	51	100	251	300	0.45093	0.17143	0.38476	0.26801	0.4281	0.1451	0.37781	0.27239
13	101	150	1	50	0.31607	5.8076	0.21355	52.554	0.2322	44.147	0.22656	42.147
14	***101***	***150***	***51***	***100***	***0.29954***	***31.962***	***0.19948***	***90.116***	***0.21792***	***57.519***	***0.2132***	***71.612***
15	***101***	***150***	***101***	***150***	***0.29219***	***48.188***	***0.19208***	***171.58***	***0.2114***	***135.96***	***0.20367***	***154***
16	***101***	***150***	***151***	***200***	***0.2646***	***50.764***	***0.16705***	***134.08***	***0.19386***	***90.476***	***0.18203***	***118.49***
17	101	150	201	250	0.30584	18.546	0.24364	49.318	0.30325	29.852	0.25069	43.421
18	101	150	251	300	0.47275	0.26082	0.42262	0.45981	0.4802	0.2702	0.43067	0.41733
19	151	200	1	50	0.38193	2.4163	0.32077	3.747	0.35694	1.4843	0.32784	3.3561
20	151	200	51	100	0.4088	0.095102	0.32114	0.87568	0.33529	0.80796	0.32327	0.8284
21	151	200	101	150	0.37253	0.22653	0.2801	1.3925	0.2939	1.2329	0.28104	1.3711
22	151	200	151	200	0.42539	0.073673	0.31561	1.3961	0.32448	1.3386	0.31156	1.3282
23	151	200	201	250	0.22828	48.116	0.19393	53.1	0.26798	3.7524	0.19778	49.633
24	151	200	251	300	0.52383	0.15612	0.48039	0.2045	0.5277	0.083878	0.48225	0.19992

The experimental values in Table 2 demonstrate the values of coordinates and texture features extracted at 0, 45, 90 and 130 degree orientations of all 24 blocks shown in figure 3e. As per figure 3e only 9 blocks numbered 3, 7-11, and 14-16 are text blocks and remaining blocks are non text blocks. The classification phase recognizes 6 blocks numbered 3, 7-10 and 16 as text blocks as their *homogeneity* and *contrast* values satisfy discriminant functions given in equations 10 and 11. And their coordinate values are recorded into feature vector *B* as shown in Table 3. The blocks 11, 14, and 15 which contain small text does not satisfy discriminant functions and are not recognized as text blocks. But they are handled properly during merging and post processing phases of the proposed method to improve the system performance.

TABLE 3: Vector *B* showing coordinate values of identified text blocks

r_{min}	r_{max}	c_{min}	c_{max}
1	***50***	***101***	***150***
51	***100***	***1***	***50***
51	***100***	***51***	***100***
51	***100***	***101***	***150***
51	***100***	***151***	***200***
101	***150***	***151***	***200***

The methodology merges identified text blocks and detects a single text region whose coordinates are stored into vector *R* as shown below. The detected text region also covers blocks 14 and 15 during merging which were undetected after classification phase thus improving detection accuracy. And the corresponding extracted text region of the image is shown in figure 3c.

$$R = [1\ 150\ 1\ 200]$$

$$\text{Where } r_{min} = 1,\quad r_{max} = 150$$

$$c_{min} = 1,\quad c_{max} = 200$$

The extracted text region now undergoes post processing phase, where during step1, the adjacent blocks 11 and 17 in right column are combined with the detected text region as their average *contrast* values satisfy the first condition in *step 3.2* of *algorithm2.* It is also noted that the non text block 17 is falsy accepted. And during step2, the adjacent unprocessed areas are not combined as they do not contain text information. Hence, the final refined text region coordinates are now shown below;

$$R = [1\ 150\ 1\ 250]$$
$$\text{Where } r_{min} = 1, \quad r_{max} = 150$$
$$c_{min} = 1, \quad c_{max} = 250$$

The performance of the system indicating detection rate before and after post processing is described in Table 4.

TABLE 4: System Performance showing detection rate before and after post processing image in figure 3a

Total no of blocks tested(# of text blocks)	Correctly detected text blocks before/ after post processing	Falsy detected text blocks before / after post processing	Missed text blocks before / after post processing	Detection rate in % before/ after post processing
24 (9)	6 / 9	0 / 01	03 / 00	66.6 / 100

The output of the system described in Table4 brings out the following details after post processing;

Detection Rate = (Number of text blocks correctly detected/ Number of text blocks tested) * 100
= (9/9) * 100 = 100%

False Acceptance Rate (FAR) = (Number of text blocks falsy detected/ Number of text blocks tested) * 100
= (01/09)*100 = 11%

False Rejection Rate (FRR) = (Number of missed text blocks / Number of text blocks tested) * 100
= (00/09)*100 = 0%

4.2 Text Region Extraction: An experimental analysis dealing with various issues

The text detection and extraction results of testing several different low resolution natural scene display board images dealing with various issues are shown in figures 4-5. And the corresponding detailed analysis is presented in Table5.

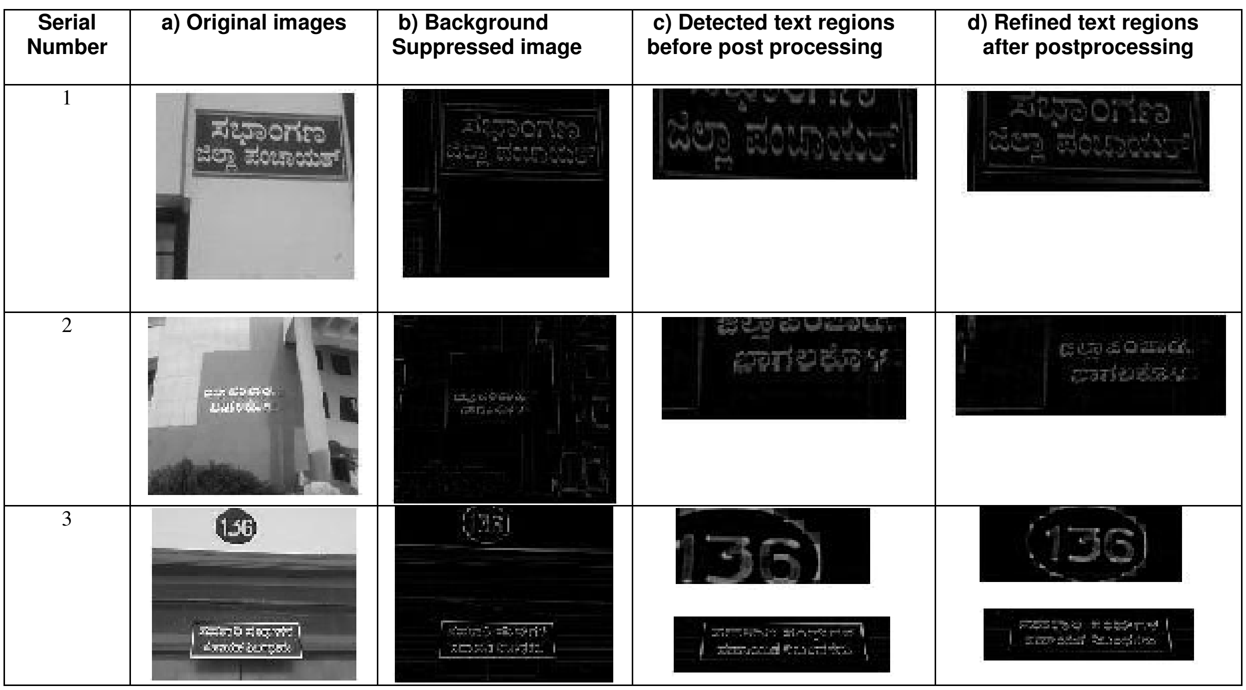

FIGURE 4: Text Extraction results of processing 3 low resolution natural scene images dealing with complex backgrounds and multiple text regions

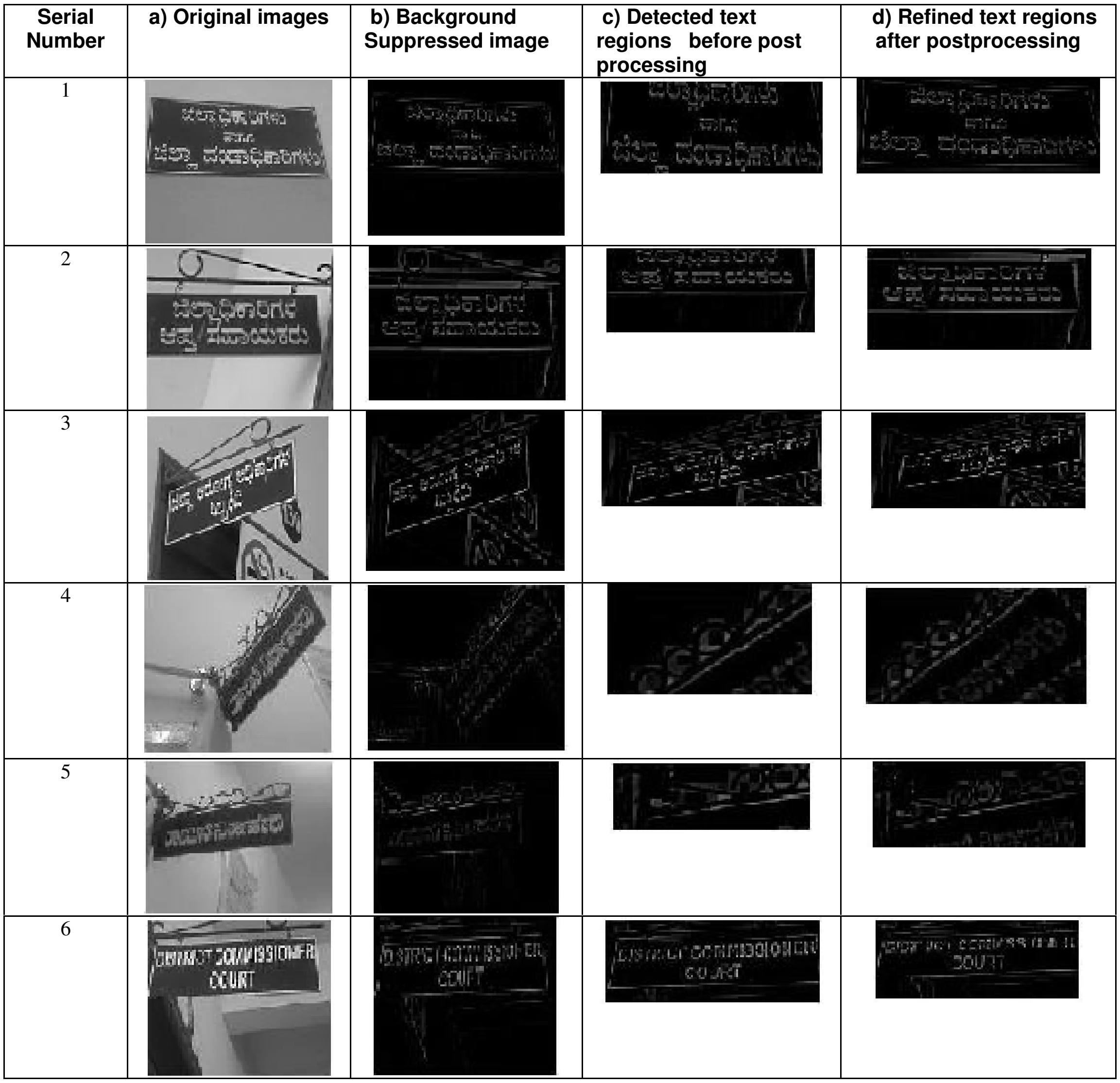

FIGURE 5: Text Extraction results of processing 6 low resolution natural scene images dealing with various issues

TABLE 5: The performance of the system of processing different images given in figures 4-5 dealing with various issues

Input Image	Total no of blocks tested(# of text blocks)	Correctly detected text blocks	Falsy detected text blocks	Missed text blocks	Detection rate (After post processing) %	Description
Figure 4-1a	24(10)	10	00	00	100	Extraction of smaller and bigger font text and elimination of unwanted background.
Figure 4-2a	24(7)	07	03	00	100	Robustness of the method in detecting correct text region by processing an image, containing text and background information, such as building walls, windows, trees and pillars etc., and uneven lighting conditions
Figure 4-3a	24(6)	06	00	00	100	Detecting multiple text regions by processing

						an image, containing complex background and uneven lighting conditions. The area containing numerals is also detected as text region.
Figure 5-1a	24(15)	15	00	00	100	Text extraction of processing natural scene image containing text of varying size, font, and alignment with varying background.
Figure 5-2a	24(12)	12	00	00	100	
Figure 5-3a	24(9)	9	00	00	100	
Figure 5-4a	24(07)	04	04	03	57.14	Text extraction result of processing an image containing non linear text and complex background
Figure 5-5a	24(08)	05	07	03	62.50	Detection of non linear text containing complex background.
Figure 5-6a	24(13)	13	00	00	100	Extraction of English text from an image with complex background

The proposed methodology has produced good results for natural scene images containing text of different size, font, and alignment with varying background. The approach also detects nonlinear text regions. Hence, the proposed method is robust and achieves an overall detection rate of 96.6%, and a false reject rate of 3.4% is obtained for 100 low resolution display board natural scene images. The method is advantageous as it uses only two texture features for text extraction. The advantage lies in less computation involved in feature extraction and classification phases of the method. The reason for false reject rate is the low contrast energy of blocks containing minute part of the text, which is too weak for acceptance to classify blocks as text blocks. However, the method has detected larger region than the actual size of the text region, when display board images with more complex backgrounds containing trees, buildings and vehicles are tested. One such result of an example is shown in 3rd row of figure 5. The system is developed in MATLAB and evaluated for 100 low resolution natural scene images on Intel Celeron (1.4GHz) computer. And it was observed that the processing time lies in the range of 6 to 10 seconds due to varying background.

As the texture features such as *homogeneity* and *contrast* used in the method does not capture language dependent information, the method can be extended for text localization from the images of other languages with little modifications. To explore such possibilities the performance of the method has been tested for localizing English text without any modifications as illustrated in 6th row of figure 5. But, the thorough experimentation is not carried out for various images containing English and other language text. The use of different block sizes can also be experimented to improve the detection accuracy and reduce false acceptance rate. The overall performance of the system of testing 100 low resolution natural scene display board images dealing with various issues is given in Table 6. The system performance is also pictorially depicted in figure 6.

TABLE 6: Overall System Performance

Total no of blocks tested(# of text blocks)	**Correctly detected text blocks**	**Falsy detected text blocks (FAR)**	**Missed text blocks (FRR)**	**Detection rate %**
2400 (1211)	1169	19 (1.5%)	42 (3.4%)	96.6 %

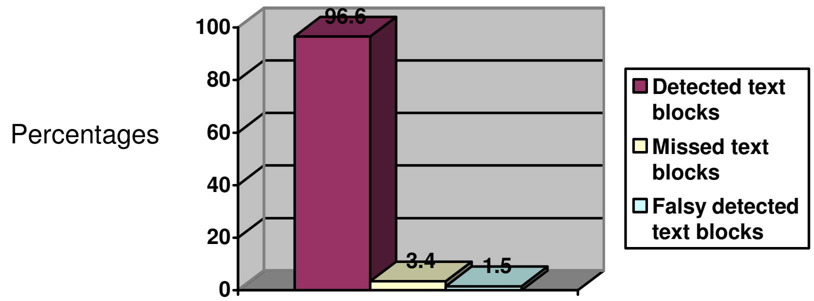

FIGURE 6: Overall results of proposed model for text localization

5. CONCLUSIONS AND FUTURE WORK

The effectiveness of the method that uses texture analysis for text localization from low resolution natural scene display board images is presented. The texture features *homogeneity* and *contrast* have performed well in detection and segmentation of text region and are the ideal choice for degraded noisy natural scene images, where the connected component analysis techniques are found to be inefficient. The intelligent post processing method has improved detection accuracy to a greater extent. The proposed method is robust and has achieved a detection rate of 96.6% on a variety of 100 low resolution natural scene images each of size 240x320.

The proposed methodology has produced good results for natural scene images containing text of different size, font, and alignment with varying background. The approach also detects nonlinear text regions. However, it detects larger text regions than the actual size when the background in the image is more complex containing trees, vehicles, and other details from sources of outdoor scenes for some images. The system is developed in MATLAB and evaluated for 100 low resolution natural scene images on Intel Celeron (1.4GHz) computer. And it was observed that the processing time lies in the range of 6 to 10 seconds due to varying background.

As the texture features such as *homogeneity* and *contrast* used in the method does not capture language dependent information, the method can be extended for text localization from the images of other languages with little modifications. The performance of the method has been tested for localizing English text, but needs further exploration.

6. REFERENCES

1. Tollmar K. Yeh T. and Darrell T. “IDeixis - Image-Based Deixis for Finding Location-Based Information”, In Proceedings of Conference on Human Factors in Computing Systems (CHI’04), pp.781-782, 2004.
2. Natalia Marmasse and Chris Schamandt. “Location aware information delivery with *comMotion*”, In Proceedings of Conference on Human Factors in Computing Systems, pp.157-171, 2000.
3. Eve Bertucci, Maurizio Pilu and Majid Mirmehdi. "Text Selection by Structured Light Marking for Hand-held Cameras" Seventh International Conference on Document Analysis and Recognition (ICDAR'03), pp.555-559, August 2003.
4. Tom yeh, Kristen Grauman, and K. Tollmar. “A picture is worth a thousand keywords: image-based object search on a mobile platform”, In Proceedings of Conference on Human Factors in Computing Systems, pp.2025-2028, 2005.

5. Abowd Gregory D. Christopher G. Atkeson, Jason Hong, Sue Long, Rob Kooper, and Mike Pinkerton, "CyberGuide: Amobile context-aware tour guide", Wireless Networks, 3(5):421-433, 1997.
6. Fan X. Xie X. Li Z. Li M. and Ma. "Photo-to-search: using multimodal queries to search web from mobile phones", In proceedings of 7th ACM SIGMM international workshop on multimedia information retrieval, 2005.
7. Lim Joo Hwee, Jean Pierre Chevallet and Sihem Nouarah Merah, "SnapToTell: Ubiquitous information access from camera", Mobile human computer interaction with mobile devices and services, Glasgow, Scotland, 2005.
8. Yu Zhong, Hongjiang Zhang, and Anil. K. Jain. "Automatic caption localization in compressed video", IEEE transactions on Pattern Analysis and Machine Intelligence, 22(4):385-389, April 2000.
9. Yu Zhong, Kalle Karu, and Anil. K. Jain. "Locating Text in Complex Color Images", Pattern Recognition, 28(10):1523-1535, 1995.
10. S. H. Park, K. I. Kim, K. Jung, and H. J. Kim. "Locating Car License Plates using Neural Networks", IEE Electronics Letters, 35(17):1475-1477, 1999.
11. V. Wu, R. Manmatha, and E. M. Riseman. "TextFinder: An Automatic System to Detect and Recognize Text in Images", IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(11):1224-1229, 1999.
12. V. Wu, R. Manmatha, and E. R. Riseman. "Finding Text in Images", In Proceedings of ACM International Conference on Digital Libraries, pp. 1-10, 1997.
13. B. Sin, S. Kim, and B. Cho. "Locating Characters in Scene Images using Frequency Features", In proceedings of International Conference on Pattern Recognition, Vol.3:489-492, 2002.
14. W. Mao, F. Chung, K. Lanm, and W. Siu. "Hybrid Chinese/English Text Detection in Images and Video Frames", In Proceedings of International Conference on Pattern Recognition, Vol.3:1015-1018, 2002.
15. A. K. Jain, and K. Karu. "Learning Texture Discrimination Masks", IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(2):195-205, 1996.
16. K. Jung. "Neural network-based Text Location in Color Images", Pattern Recognition Letters, 22(14):1503-1515, December 2001.
17. K. Y. Jeong, K. Jung, E. Y. Kim and H. J. Kim. "Neural Network-based Text Location for News Video Indexing", In Proceedings of IEEE International Conference on Image Processing, Vol.3:319-323, January 2000.
18. H. Li. D. Doerman and O. Kia. "Automatic Text Detection and Tracking in Digital Video", IEEE Transactions on Image Processing, 9(1):147-156, January 2000.
19. H. Li and D. Doermann. "A Video Text Detection System based on Automated Training", In Proceedings of IEEE International Conference on Pattern Recognition, pp.223-226, 2000.
20. W. Y. Liu, and D. Dori. "A Proposed Scheme for Performance Evaluation of Graphics/Text Separation Algorithm", Graphics Recognition – Algorithms and Systems, K. Tombre and A. Chhabra (eds.), Lecture Notes in Computer Science, Vol.1389:359-371, 1998.
21. Y. Watanabe, Y. Okada, Y. B. Kim and T. Takeda. "Translation Camera", In Proceedings of International Conference on Pattern Recognition, Vol.1: 613-617, 1998.
22. I. Haritaoglu. "Scene Text Extraction and Translation for Handheld Devices", In Proceedings IEEE Conference on Computer Vision and Pattern Recognition, Vol.2:408-413, 2001.
23. K. Jung, K. Kim, T. Kurata, M. Kourogi, and J. Han. "Text Scanner with Text Detection Technology on Image Sequence", In Proceedings of International Conference on Pattern Recognition, Vol.3:473-476, 2002.
24. H. Hase, T. Shinokawa, M. Yoneda, and C. Y. Suen. "Character String Extraction from Color Documents", Pattern Recognition, 34(7):1349-1365, 2001.
25. Ceiline Thillou, Silvio Frreira and Bernard Gosselin. "An embedded application for degraded text recognition", EURASIP Journal on applied signal processing, 1(1):2127-2135, 2005.

26. Nobuo Ezaki, Marius Bulacu and Lambert Schomaker. “Text Detection from Natural Scene Images: Towards a System for Visually Impaired Persons”, In Proceedings of 17th International Conference on Pattern Recognition (ICPR’04), IEEE Computer Society vol.2:683-686, 2004.
27. Chitrakala Gopalan and Manjula. “Text Region Segmentation from Heterogeneous Images”, International Journal of Computer Science and Network Security, 8(10):108-113, October 2008.
28. Rui Wu, Jianhua Huang, Xianglong Tang and Jiafeng Liu. ” A Text Image Segmentation Method Based on Spectral Clustering”, IEEE Sixth Indian Conference on Computer Vision, Graphics & Image Processing, 1(4): 9-15, 2008.
29. Mohammad Osiur Rahman, Fouzia Asharf Mousumi, Edgar Scavino, Aini Hussain, and Hassan Basri. ”Real Time Road Sign Recognition System Using Artificial Neural Networks for Bengali Textual Information Box”, European Journal of Scientific Research, 25(3):478-487, 2009.
30. Rajiv K. Sharma and Amardeep Singh. “Segmentation of Handwritten Text in Gurmukhi Script”, International Journal of Image Processing, 2(3):13-17, May/June 2008.
31. Amjad Rehman Khan and Zulkifli Mohammad. ”A Simple Segmentation Approach for Unconstrained Cursive Handwritten Words in Conjunction with the Neural Network”, International Journal of Image Processing, 2(3):29-35, May/June 2008.
32. Amjad Rehman Khan and Zulkifli Mohammad. ”Text Localization in Low Resolution Camera”, International Journal of Image Processing, 1(2):78-86, 2007.
33. R.M. Haralick, K. Shanmugam and I. Dinstein. “Textural Features for Image Classification”, IEEE Transactions Systems, Man, and Cyber-netics, 3(6):610-621, 1973.