Research Journal of Applied Sciences, Engineering and Technology 8(4): 556-564, 2014
DOI:10.19026/rjaset.8.1005
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: April 22, 2014 Accepted: July 01, 2014 Published: July 25, 2014

Research Article

Separation of Text Components from Complex Colored Images

G. Gayathri Devi and C.P. Sumathi
Department of Computer Science, SDNB Vaishnav College for Women, Chennai, Tamil Nadu, India

Abstract: The objective of this study is to project a new methodology for text separation in an image. Gamma Correction Method is applied as a preprocess technique to suppress non text regions and retain text regions. Text Segmentation is achieved by applying Positional Connected Component Labeling, Text Region Extraction, Text Line Separation, Separation of Touching Text and Separation of Text Components algorithms. At last, the details of each word's and the line's starting text component position are stored in a text file. Experiments are conducted on various images from the datasets collected and tagged by the ICDAR Robust Reading Dataset Collection Team. It is observed that the proposed method has an average recall rate of 97.5% on separation of text components in an image.

Keywords: Connected components, gamma correction method, segmentation, text extraction, text line, text separation, thinning

INTRODUCTION

Rapid development of digital technology has resulted in digitization of all categories of materials. Text data present in images and video contain useful information for detection of vehicle license plate, name plates, keyword based image search, content based retrieval, text based video indexing, video content analysis, document retrieving, address block location etc. Recognition of the text data in document images depends on the efficient separation of text. Many methods have been proposed for text separation in images and videos. It is not easy to describe a unified method as there are low-contrast or complex images, text with variations in font size, style, color, orientation and alignment etc.

LITERATURE REVIEW

Chethan and Kumar (2010) proposed an algorithm to remove graphics from the document and correct skew for the documents captured using cellular phone. The basic process of this approach consists of three steps: First, a vertical and horizontal projection was used to remove graphics from images. Second, dilation operation was applied to the binary Images and the dilated Image was thinned. At last Hough transform was applied to skew angle.

A new text line location and separation algorithm for complex handwritten documents was proposed by Shi and Govindaraju (2004). The method used a concept of fuzzy directional run length which imitated an extended running path through a pixel of a document. The method partitioned the complex documents to separate the content of the document to texts in terms of text words or text lines and to other graphic areas.

Peng *et al*. (2013) projected a method to classify machine printed text, handwritten text and overlapped text. Three different classes were initially identified using G-means based classification followed by a Markov Random Field (MRF) based relabeling procedure. A MRF based classification approach was then used to separate overlapped text into machine printed text and handwritten text using pixel level features.

Patil and Begum (2012) presented a method for discriminating handwritten and printed text from document images based on shape features. K-nearest neighbor based on minimum distance was used to classify the handwritten and printed text words.

Yao *et al*. (2012) proposed a system which detected texts of arbitrary orientations in natural images. The proposed algorithm consists of four stages: component extraction where image are grouped together to form connected components using a simple association rule, component analysis to remove non-text parts, candidate linking to link the adjacent character candidates into pairs and chain analysis to discard the chains with low classification scores. Coates *et al*. (2011) presented text detection and recognition system based on scalable feature learning algorithm and applied it to images of text in natural scenes.

Corresponding Author: G. Gayathri Devi, Department of Computer Science, SDNB Vaishnav College for Women, Chennai, Tamil Nadu, India

A new method to locate text in images with complex background had been presented by Gonzalez method mainly composed of three main stages: a segmentation stage to find character candidates, a connected component analysis based on fast-to-compute but robust features to accept characters and discard non-text objects and finally a text line classifier based on gradient features and support vector machines.

Phan *et al*. (2012) proposed novel symmetry features for text detection in natural scene images. Within a text line, the intra-character symmetry captured the correspondence between the inner contour and the outer contour of a character while the inter-character symmetry helped to extract information from the gap region between two consecutive characters. A formulation based on Gradient Vector Flow was used to detect both types of symmetry points. These points were then grouped into text lines using the consistency in sizes, colors and stroke and gap thickness.

A real-time scene text localization and recognition method was presented by Neumann and Matas (2012) The probability of each of Extremal Regions (ER) was estimated using novel features calculated with O (1) complexity and only ERs with locally maximal probability are selected to classify into character and non-character classes using SVM classifier with the RBF kernel.

A top-down, projection-profile based algorithm to separate text blocks from image blocks in a Devanagari document was proposed by Khedekar *et al*. (2003). They analyzed the pattern produced by Devanagari text in the horizontal corresponding to a text block possesses certain regularity in frequency, orientation and shows spatial cohesion.

PROPOSED METHODOLOGY

The aim of the proposed work is to separate the text component from the input Image. The work flow of the system is shown in the Fig. 1. The input image of the proposed system has a complex background with text in it. The first stage is pre-processing that suppresses the non-text background details from the image by applying appropriate gamma value. Otsu's thresholding algorithm is used to calculate the threshold value and applied to this image to create an output binary image.

et al. (2012). The method combined efficiently MSER and a locally adaptive thresholding method. The

The output may contain white and black text region and some noises. In next stage of Text Region Extraction algorithm, white text and black text region are extracted from the binary image and those text regions are stored as white foreground in black background. Also, the algorithm removes very small and large non text regions from the image.

In next stage the binary text image is used to create text row images. Adjacent Text component in Text Row image may touch each other and overlap or touching of text components of consecutive row images may exist. The Separation of Text Row (STR) Algorithm is used to break the text binary image in to row text images and solve the problem of touching components of consecutive row images. The separation of the touching component of same text row image is done by applying Detect and Split Touching Text (DSTT) Algorithm.

The aim of next stage is used to determine the individual text component, words and lines of a text row image. Sequence Separation of Text Components (SSTC) Algorithm is used to separate the individual text component from the row Images. Text position Details Algorithm saves the details of the Text Position of each Text Components, Row Line and Words are stored in text files. Positional Connected Component Labeling (PCCL) algorithm which finds the connected component is used in stages of text separation.

Preprocess technique by using Gamma Correction Method (GCM): The Gamma Correction method proposed by Sumathi and Devi (2014) suppresses the non-text background details from the image by applying appropriate gamma value and to remove non text region. The algorithm estimated the Gamma Value (GV) without any prior details of the imaging device by using texture measures. By applying this estimated gamma value to an input image (Fig 2a, c, e), the background suppressed image (Fig. 2b, d, f) will be achieved. Otsu's thresholding algorithm is used to calculate the threshold value and applied to this image to create an output binary image (Fig. 3a, 4a, 5a). This binary Image (I) will be the input of the Extraction of Text algorithm.

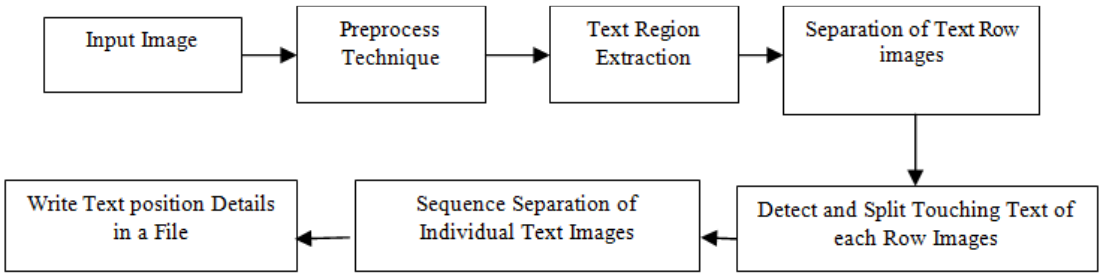


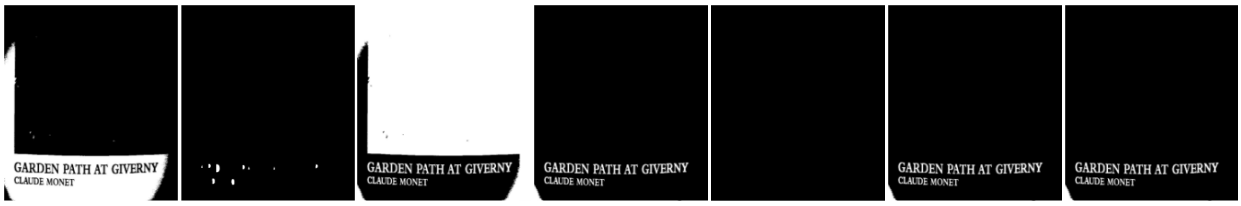
Fig. 1: Work flow of text separation from image

(a) (b) (c) (d) (e) (f)

Fig. 2: (a) Image 1, (b) gamma corrected image of Fig. 1a (GV = 0.7), (c) image 2, (d) gamma corrected image Fig. 1c (GV = 7), (e) image 3, (f) gamma corrected image Fig. 1e (GV = 5.5)

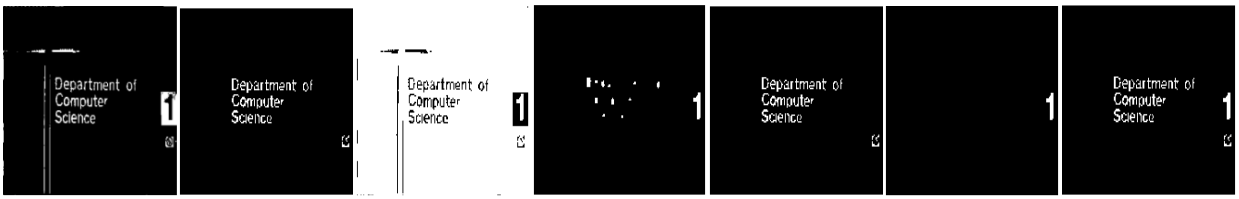
(a) (b) (c) (d) (e) (f) (g)

Fig. 3: (a) Input image (I), (b) White connected components of I, (c) Reversed Image (RI), (d) white components of reversed image, (e) white text components of I, (f) black text components of I, (g) output



(a) (b) (c) (d) (e) (f) (g)

Fig. 4: (a) Input image (I), (b) white connected components of I, (c) Reversed Image (RI), (d) white components of reversed image, (e) white text components of I, (f) black text components of I, (g) output



(a) (b) (c) (d) (e) (f) (g)

Fig. 5: (a) Input image (I), (b) white connected components of I, (c) Reversed Image (RI), (d) white components of reversed image, (e) white text components of I, (f) black text components of I, (g) output

Positional connected component labeling algorithm: A set of pixels in which each pixel is connected to all other pixels is called a connected Component. A component labeling algorithm finds all connected components in an image and assigns a unique label to all points in the same component. The Positional Connected Component Labeling (PCCL) algorithm proposed by Devi and Sumathi (2014) is based on 8-connectivity to find all connected components in an image, assigns an unique label to all points in the same component and find number of components present in the image. The algorithm for PCCL is based on the position of the white pixels in the image.

Positional Connected Component Labeling (PCCL) algorithm:

Input: Binary Image Matrix (I)

Output: Connected Component Labeled Matrix (L), Numbers_of_Components

Step 1: Find the foreground (white) pixel and record the foreground column position of the binary matrix image in a matrix (Position Matrix).

Step 2: Unmark all the cells of the Position Matrix (PM).

Step 3: Find the minimum position value (MinPos) and maximum position value (MaxPos) from the Position Matrix.
Step 4: Set the value of LABEL to 1. Current Row (CR) to 1.AV [] = {}, PV [] = {}
Step 5: Get the first unmarked value (umv) from the Position Matrix (PV [] = umv). Set CR to the Row Number of umv. FLAG = NOTPREVMARKED and L = LABEL. If no unmarked cell found, then go to step 15.
Step 6: Find out the adjacent value (AV []) of the position values (PV []) from PM. The adjacent values are P-1, P and P+1 for a value P. If P is to be MinPos, then the adjacent values are P and P+1. If P is to be MaxPos, then the adjacent values are P-1 and P.
Step 7: Search for the AV [] in the CR, CR-1 and CR+1. Mark the corresponding cell If any of these cells are already Labeled by the previous pass, change the FLAG = PREVMARKED, L = Label assigned to the already Labeled cell. (Do not include CR-1 for the first row and CR+1 for the last row).
Step 8: Increment CR by one.
Step 9: Scan CR and find the adjacent values (AV []) of cells marked by step 7 in the row CR.
Step 10: PV [] = AV []. Go to step 6 if AV [] = Ø or CR>LAST ROW.
Step 11: Assign L value for the corresponding cells marked during this pass in the Input Image (I).
Step 12: If FLAG IS NOTPREVMARKED then increment LABEL value by one.
Step 13: If any unmarked cells found go to step 5.
Step 14: Number_of_Components = LABEL - 1.
Step 15: Stop the procedure.

Text Region Extraction (ETR) algorithm: The Binary image obtained by previous phase may contain white and black text region. In Text Region Extraction algorithm white text region in black background and black text region in white background are extracted from the binary image and those text regions are stored as white foreground in black background. The algorithm uses PCCL Algorithm to find connected components. Also, the algorithm removes very small and large components from the image.

The algorithm for extraction of text region is presented as follows:

Input: Binary Image Matrix I
Output: Text Image Matrix L, No_of_Text_ Components

Step 1: Apply PCCL Algorithm for the Input Image I to produce Image L1. This algorithm treat white pixel as foreground (Region of Interest).
Step 2: Find the size of each component and remove very small and large component from the Image L1. (White Text Components of I are extracted).
Step 3: Reverse the Image I (Change the white pixel to black pixel and black pixel to white pixel). RI = ~I.
Step 4: Apply PCCL Algorithm for the Reversed Image RI to produce Image L2.
Step 5: Find the size of each component and remove very small and large Component from the Image L2. (Black Text Components of I are extracted.)
Step 6: Find all the Components (RCs) of L1 that fit inside the components of L2. Remove RCs from the image L1.
Step 7: Find all the Components (RCs) of L2 that fit inside the component of L1. Remove RCs from the image L2.
Step 8: Assign 1 to the pixel value greater than zero for the image L1 and L2. (L1 = (L1>0), L2 = (L2>0)).
Step 9: L = L1+L2. Apply PCCL Algorithm for the L to assign the label value.
Step 10: No_of_Text_ Component = No_of_ Components.
Step 11: Stop the procedure.

To illustrate the method Fig. 3a is taken. Region of Interests are G, O (white pixel) T, e, s, c, o, D, i, e, t, s, l, o, s, e, a, s, t, o, n, e, i, n, 8, w, e, e, k, s (Black pixel). The output after step 2 of ETR algorithm is shown in Fig. 3b. As per step 3, Fig. 3a is reversed (Fig. 3c) to find out whether there is any black region of interest component. There are 6 components which looks like filled ‘o’ in Fig. 3c and one component look like filled ‘o’ near right bottom corner in Fig. 3d. White pixels inside the letter ‘e’ in Tesco, letters ‘o’ in lose, ‘a’, letter ‘o’ in stone and digit ‘8’ are treated as components by Positional Connected Component Algorithm. However the white pixel inside the letter ‘e’ is also a component. But, according to step 2 of ETR algorithm, it is treated as very small component and it has been removed. These 6 components appeared in Fig. 3b and one component in Fig. 3d, 4d and 5d is removed after step 6 and 7 of ETR algorithm. The output of step 6 of ETR is Fig. 3e, 4e and 5e and the output of step 7 is Fig. 3f. Merge the output of step 6 and 7 of ETR algorithm to get the output image L (Fig. 3g). Figure 4b, f and Fig. 5b, f are the stages involved when the algorithm is applied for the Fig. 4a and c, respectively.

Separation of Text Row (STR) algorithm: The aim of this algorithm is to break the image in to row images by using maximum and minimum row position of the text components.

Separation of text line extraction method consists of the following steps:

Fig. 6: (a) Temp row 1, (b) row 1, (c) temp row 2, (d) row 2, (e) temp row 3, (f) row 3, (g) temp row 4, (h) row 4, (i) temp row 5, (j) row 5, (k) temp row 6, (l) row 6

Fig. 7: Row 1 of Fig. 4g

Input: Labeled Text Image L, No_of_Text Components

Ouput: Row Images R [], No_of_Row

Step 1: Extract each component and store it in CompDet []
Step 2: Reassign the label of Text Components in Image L in the sequence order according to Top Row Pixel Position of each component
Step 3: StartLabel = 1, End Label = No_Of_ Text_Components, No_Of_Row = 0
Step 4: While StartLabel< = EndLabel

i. Find the Minimum Row Position (MinRowPos) and Maximum Row Position (MaxRowPos) of the Component Labeled as StartLabel
ii. Components lies between (1, MinRowPos, MaxCol, MaxRowpos) in image L forms a text Row images TempRow
iii. EndLabel = Max Label of TempRow image
iv. Store Minimum Row Position of the component among the components of Temp Row in to MinY
v. Store Maximum Row Position of the component among the components of Temp Row in to MaxY
vi. RI = L (1, MinY, MaxCol, MaxY)
vii. RI = 0 for the values other than (RI> = StartLabel and RI< = EndLabel)
viii. R [++No_Of_Row] = RI
ix. StartLabel = EndLabel+1

Step 5: Stop the Procedure

To illustrate the STR algorithm, Fig. 3g, 4g and 5g is taken. The output of step 4 (ii) of STR algorithm is shown in the Fig. 6a, c, e, g, i and k. In Fig. 6g the component 'e' in word 'stone' is the StartLabel as the TopRowPosition is greater than the other component. The exact row height of component 'e' from column 1 to maximum column of image is examined. Some partial part of l, o, s, e, a, s, t components are found in Fig. 6g and decided that those components are also belongs to the same row. Figure 6h is obtained after step 4 (vi) of ETL is applied. The Output Rows R [] are shown in Fig. 6b, d, f, h, j and i. In Fig. 6i a part of 'l', 'o', 's' components appears, but those components will not be taken in to consideration in creation of new row as those components already been found in previous row (Step 5 (vii) of STR) (Fig. 7).

Detect and Split Touching Text (DSTT) algorithm: This algorithm uses component width size, Outlier algorithm to detect the touching components. The separation of the touching component is done by using morphological thinning and lightly populated area algorithm.

The DSTT algorithm is presented by the following steps.

Input: Row Images R [], No_Of_Row
Output: Corrected Row Images R []

Step 1: Repeat step 2 to step 5 For i = 1 to No_OF_Rows
Step 2: Find the component width size of each component of row R [i] and store it in CWS []
Step 3: Find the outlier value (s) of CWS []
Step 4: Calculate the average value (avg) of CWS [] except outlier values (s) (Number greater than Q3+1.5×IQR is an Outlier. Q3 Third Quartile, Q = First Quartile, IQR = Q3 - Q1)
Step 5: Excepted_Component_size = avg+avg/6
Step 6: For N = 1 to No_of_Components of R [i]

a. If (CWS (N) > = Excepted_Component_size)
i. Morphological Thinning Algorithm is applied on Text Component
ii. Check for the Component Width Size of new ones
iii. If it does not lie below ECS then
1. Find the Junction Point [] of the lightly populated white pixels approximately around ECS
2. Split the component in to n component at the Junction Point []

Step 7: Stop the Procedure

To illustrate the DSTT algorithm, Fig. 7 is taken as the input. CsWS [] = {51, 165, 44, 57, 44, 54, 50, 59,

(a) (b) (c) (d) (e) (f) (g)

Fig. 8: Detect and split of touching component 1

(a) (b) (c) (d) (e)

Fig. 9: Detect and split of touching component 2

(a) (b) (c) (d) (e) (f) (g)

Fig. 10: Detect and split of touching component 3

Fig. 11: Corrected row 1 of Fig. 4g

53, 49, 51, 122, 161} is obtained after step 2 is executed. The outliers values calculated as per algorithm are 161 and 166. The average (avg) value of all CWS [] except Outlier Value (s) is 57.63. The Expected Component Size ECS is 69. The size of the component of Fig. 8a is 165 and it is greater than 69. So, Step 6 (i) is executed. The output Fig. 8b is shown after Morphological Thinning is applied. The two new components are shown in Fig. 8c and d. The size of component in Fig. 8c is greater than Excepted_ Component_size. So, the Component is split at the calculated junction points (Fig. 8f). Now, the Component is split in to 2 components (Fig. 8g and d). Figure 8 to 10 are obtained on 2st, 3nd, 4th iteration of step 6, respectively. The output of the algorithm is shown in Fig. 11. The algorithm is insensitive to the size of the Text as Component Width Size is calculated for each row.

Sequence Separation of Text Components (SSTC) algorithm: This algorithm is used to separate the individual text component from the row Images. Components of each text line obtained by text row algorithm are sorted in ascending order according to the left most column position of the component. The components are extracted one by one according to the sequence order. The SSTC algorithm is formulated by the following steps.

Input: Row Images R []
Output: Corrected Row Images R []-Label is assigned sequentially from the first component of first Row to the Last Component of Last Row, NewLineNo [], Total_No_of_Component, TextComponents []

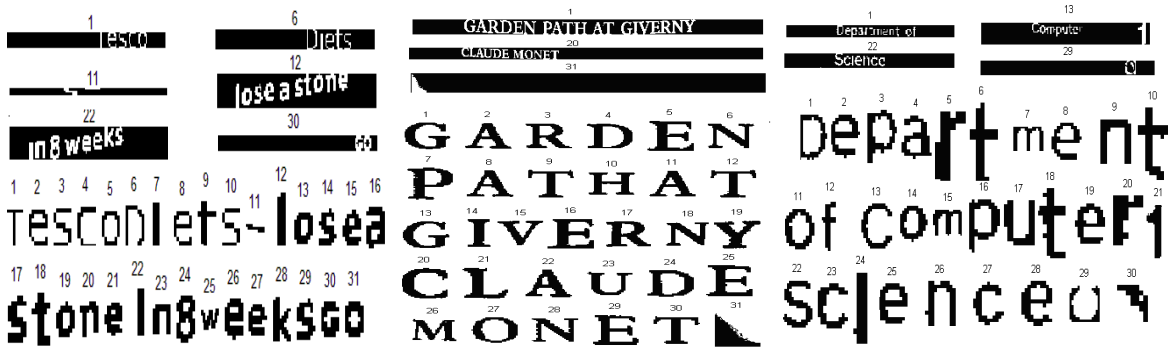
Step 1: StartLabel = 1
Step 2: Repeat Step 3 to 7 for i =1 to No_of_Row
Step 3: PCCL Algorithm is applied on R [i] as there may be an increase of number of components due to detect and Split Touching Text Algorithm
Step 4: Reassign the label of Text Components in Row R [i] in the sequence order starting from StartLabel according to Left Column Pixel Position of each component
Step 5: NewLineNo [i] = MinLabel assigned in the Row R [i]
Step 6: Total_No_of_Component = MaxLabel assigned in the Row R [i]
Step 7: For j = StartLabel to Total_No_of_Component

i. [r c] = find (R [i] = = j)
ii. TextComponents [j] = R [j] (min (c), min (r): max (c), max (r))

Step 8: StartLabel = maxLabel assigned in the Row R [i] +1
Step 9: Stop the Procedure

According to step 4 of SSTC Algorithm, Components of each text Row R [] obtained are sorted in ascending order according to the left most column position of the component in the sequence order. Text Components are shown in Fig. 12a to c. The array NewLineNo [] contains the starting text component numbers of each Row.

Text position details algorithm: The Details of the Text Position of each Text Components, Row Line and Words are stored in text files. The starting text component numbers of each Row is already obtained by the Sequence Separation of Text Components algorithm and those details are stored in 'newlinedet.txt'. Space Details of each text component are calculated to frame



(a) (b) (c)

Fig. 12: (a) Text separation 1, (b) text separation 2, (c) text separation 3

Table 1: Text row start position of Fig. 2c

Row start position	1	20	31

words. The algorithm of Text Position Detail (TPD) is given below (Table 1).

Input: Row Images R [], NewLineNo [], No_Of_Row
Output: Text Position Details

Step 1: File Open to save details of Text position

a. fidsp = fopen ('spacedet.txt','wt')
b. fidword = fopen ('worddet.txt','wt')
c. fidnl = fopen ('newlinedet.txt', 'wt')

Step 2: Starting position of component of each row are stored in newlinedet.txt using NewLineNo []
Step 3: Repeat Step 4 to Step 6 for i = 1 to No_Of_Row
Step 4: Space Gap between each successive component are calculated and stored in sp [] and in 'spacedet.txt'. (SpaceGap = -1000 for the first character of each row.)
Step 5: Outlier Value (s) of sp [] are calculated (exclude SpaceGap of -1000)
Step 6: Outlier space gaps are treated as a delimiter of a word. These details are stored in worddet.txt
Step 7: Stop the Procedure

To illustrate the example the image in Fig. 2c is taken. In step 2 of TPD algorithm, the starting text position calculated in NewLineNo [] by SSTC algorithm is saved in 'newlinedet.txt' and the component Position, left most column position of component, right most column position of component, width size of component and the gap of 2 adjacent Text Components (Gap as -1000 for the first character of new row) are stored in 'spacedet.txt'. To Find the starting position of each word, sp [] (Gap of 2 adjacent text component) for each row calculated is examined. Here the outlier of sp [] for each row is calculated and the text position whose gaps are outliers (marked in red color in Table 2 to 5 and 8) or -1000 are saved as word gap in 'worddet.txt.'

Description of C1, C2, C3, C4, C5 and N of Table 2 to 9 are given below.

C1-Text Position, C2-Min Column position of Current Text, C3-Max Column of Previous Text, C4-Text Width, C5-Gap between 2 adjacent Text (C2-C3), N = -1000.

EXPERIMENTAL RESULTS

The performance of the proposed technique has been evaluated based on Precision, Recall and F-Score Measure. Precision and Recall rates have been computed based on the number of correctly detected characters (TP) in an image, in order to evaluate the efficiency and robustness of the algorithm. The metrics are as follows.

Definition 1: False Positives (FP) /False alarms are those regions in the image which are actually not characters of a text, but have been detected by the algorithm as text.

Definition 2: False Negatives (FN) /Misses are those regions in the image which are actually text characters, but have not been detected by the algorithm.

Definition 3: Precision rate (P) is defined as the ratio of correctly detected characters to the sum of correctly detected characters plus false positives:

(Precision = TP/ (TP + FP) *100%)

Definition 4: Recall rate (R) is defined as the ratio of the correctly detected characters to sum of correctly detected characters plus false negatives:

(Recall = TP/ (TP + FN) *100%)

Definition 5: F-Score is the harmonic mean of recall and precision rates.

Table 2: Text space detail of Fig. 2c

C1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
C2	179	218	259	301	345	382	447	477	515	555	616	654	715	756	774	814
C3	217	258	300	341	378	424	479	516	552	599	655	691	753	773	816	846
C4	38	40	41	40	33	42	32	39	37	44	39	37	38	17	42	32
C5	N	1	1	1	4	4	23	-2	-1	3	17	-1	24	3	1	-2
C1	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
C2	848	889	933	179	205	229	254	285	315	356	392	424	456	482	129	
C3	887	931	969	203	228	256	284	312	338	388	420	453	479	508	174	
C4	39	42	36	24	23	27	30	27	23	32	28	29	23	26	45	
C5	2	2	2	N	2	1	-2	1	3	18	4	4	3	3	N	

Table 3: Word start position of Fig. 2c

Word start position	1	7	11	13	20	26	31

Table 4: Text row start position of Fig. 2a

Row start position	1	6	11	12	22	30

Table 5: Text space detail of Fig. 2a

C1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
C2	73	83	92	100	108	75	86	90	98	105	46	16	21	31	39	52
C3	81	89	97	105	114	82	87	96	103	110	56	18	28	37	46	58
C4	8	6	5	5	6	7	1	6	5	5	10	2	7	6	7	6
C5	N	2	3	3	3	N	4	3	2	2	N	N	3	3	2	6
C1	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
C2	64	71	79	89	99	17	22	34	46	60	69	78	87	112	119	
C3	69	76	86	96	106	19	29	41	57	66	76	85	93	117	124	
C4	5	5	7	7	7	2	7	7	11	6	7	7	6	5	5	
C5	6	2	3	3	3	N	3	5	5	3	3	2	2	N	2	

Table 6: Word start position of Fig. 2a

Word start position	1	6	11	12	16	17	22	24	25	30

Table 7: Text row start position of Fig. 2e

Row start position	1	13	22	29

Table 8: Text space detail of Fig. 2e

C1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
C2	102	119	132	145	158	167	177	195	208	220	241	254	103	118	132
C3	115	129	142	155	163	173	192	205	216	227	251	259	116	130	148
C4	13	10	10	10	5	6	15	10	8	7	10	1	13	12	16
C5	N	4	3	3	3	4	4	3	3	4	14	3	N	2	2
C1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
C2	151	164	176	185	198	315	102	117	128	133	146	158	170	317	322
C3	161	173	183	195	203	330	114	126	130	143	155	168	179	331	328
C4	10	9	7	10	5	15	12	9	2	10	9	10	9	14	6
C5	3	3	3	2	3	112	N	3	2	3	3	3	2	N	-9

Table 9: Word start position of Fig. 2e

Word Start Position	1	11	13	21	22	29

The experimentation of the algorithms implemented in MATLAB Tool was carried out on the ICDAR data set consisting of 100 different images and as well as some images were taken from the WEB. Some of the Experiments results have been shown in Section above. The results in this research show that the new proposed method separates the text component of the image. The proposed method can separate most of the text region successfully, including text with different styles, size, font, orientations and color. This approach resulted in an average precision rate of 88%, recall rate of 97.5% and F-Score of 92.5%.

CONCLUSION AND RECOMMENDATIONS

The study presents a new algorithm for the separation of text region information in an image. This proposed method uses a positional connected component labeling, text region extraction, text line separation, separation of touching text and separation of text components algorithms. The proposed technique is an essential stage for most of the object recognition method. The algorithm is applied on several images with text of different styles, size, font, alignment and complex backgrounds taken from ICDAR datasets and shown promising results. The future work concentrates on the next stage of developing a text recognition algorithm from the output obtained by the newly proposed text separation technique.

REFERENCES

Chethan, H.K. and G.H. Kumar, 2010. Graphics separation and skew correction for mobile captured documents and comparative analysis with existing methods. Int. J. Comput. Appl., 7(3): 42-47.

Coates, A., B. Carpenter, C. Case and S. Satheesh, 2011. Text detection and character recognition in scene images with unsupervised feature learning. Proceeding of International Conference on Document Analysis and Recognition (ICDAR, 2011), pp: 440-445.

Devi, G.G. and C.P. Sumathi, 2014. Positional connected component labeling algorithm. Indian J. Sci. Technol., 7(3): 306-311.

Gonzalez, A., L.M. Bergasa, J.J. Yebes and S. Bronte, 2012. Text location in complex images. Proceeding of 21st International Conference on Pattern Recognition (ICPR, 2012). Tsukuba, Japan, pp: 617-620.

Khedekar, S., V. Ramanaprasad, S. Setlur and V. Govindaraju, 2003. Text- image separation in Devanagari documents. Proceeding of the 7th International Conference on Document Analysis and Recognition, pp: 1265-1269.

Neumann, L. and J. Matas, 2012. Real-time scene text localization and recognition. Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (CVPR, 2012), pp: 3538-3545.

Patil, U. and M. Begum, 2012. Word level handwritten and printed text separation based on shape features. Int. J. Emerg. Technol. Adv. Eng., 2(4): 590-594.

Peng, X., S. Setlur, V. Govindaraju and R. Sitaram, 2013. Handwritten text separation from annotated machine printed documents using Markov Random Fields. Int. J. Doc. Anal. Recog., 16: 1-16, DOI 10.1007/s10032-011-0179.

Phan, T.Q., P. Shivakumara and C.L. Tan, 2012. Detecting text in the real world. Proceeding of the 20th ACM International Conference on Multimedia (MM '12), pp: 765-768.

Shi, Z. and V. Govindaraju, 2004. Line separation for complex document images using fuzzy runlength. Proceeding of 1st International Workshop on Document Image Analysis for Libraries, pp: 306-312.

Sumathi, C.P. and G.G. Devi, 2014. Automatic text extraction from complex colored images using gamma correction method. J. Comput. Sci., 10(4): 706-715.

Yao, C., X. Bai, W. Liu, Y. Ma and Z. Tu, 2012. Detecting texts of arbitrary orientations in natural images. Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (CVPR, 2012), pp: 1083-1090.