USE OF GAUSSIAN PYRAMID FOR MSER BASED TEXT EXTRACTION FROM SCENE IMAGE

A Thesis submitted by

ABHIDIP BHATTACHARYYA

In the partial fulfillment of the requirements for the degree of

MASTER OF TECHNOLOGY

In

COMPUTER SCIENCE

Under the guidance of

Dr. UJJWAL BHATTACHARYYA

Associate Professor (Equiv.)
Computer Vision and Pattern Recognition Unit
Indian Statistical Institute
Kolkata

INDIAN STATISTICAL INSTITUTE

KOLKATA
WEST BENGAL

JULY 2014

Declaration of Originality and Compliance of Academic Ethics

"I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text."

Name: **Abhidip Bhattacharyya**

Roll Number: **MTC1202**

Thesis Title: **Use of Gaussian Pyramid for MSER Based Text Extraction From Scene Image**

Signature:

Date:

Indian Statistical Institute

Certificate of Approval

This is to certify that the thesis entitled ***Use of Gaussian Pyramid for MSER Based Text Extraction From Scene Image*** submitted by ***Mr. Abhidip Bhattacharyya*** to Indian Statistical Institute, Kolkata towards partial fulfillment of the requirements for the award of the degree of Master of Technology in Computer Science is a bonafide record of the work carried out by him under my supervision and guidance.

Thesis Supervisor

Dr. Ujjwal Bhattacharya

Associate Professor (Equiv.)
Computer Vision and Pattern Recognition Unit
Indian Statistical Institute
Kolkata

Acknowledgements

This work would not have been possible without the continuous support and encouragement provided by my thesis supervisor, Dr. Ujjwal Bhattacharya, Associate Professor (Equiv.) at Computer Vision and Pattern Recognition Unit. His able assistance, timely suggestions and personal guidance throughout the duration of the project has played a major role without which I would have never been able to reach this far.

I take this opportunity to also thank, Dr. Bidtyut Baran Chaudhuri, Professor and Head CVPR unit, ISI, for providing me conductive ambience for developing my project.

I am grateful all other trainee, project linked personals and doctoral scholars with whom I share the lab, for their valuable help and suggestions.

At last but most importantly I would like to thank my beloved family. Without their support and inspiration I could never have achieved this golden chance of being a part of Indian Statistical Institute..

Date:_______________

(Abhidip Bhattacharyya)

Abstract

The potential of automatic extraction of texts from scene image as an application is ever increasing with the advancement of technology especially after market deluging with smart-phones. However, it is a difficult problem considering the enormous variations in lighting conditions, presence of noise etc. in such images. Researchers are now working extensively towards developing a robust strategy for this purpose. A few standard databases of camera captured scene images are now available publicly for reporting the performance of each new strategy. During the last one year we studied several strategies towards the development of a robust method for extraction of scene texts from such camera captured outdoor scenes. In this study, we developed a novel scheme for scene text extraction using Gaussian pyramid decomposition of input image and obtaining Maximally Stable Extremal Regions (MSERs) at each level of the Gaussian pyramid to use information at different scales. We select only a subset of MSERs at each level based on a few commonly used rules. We carefully decided a set of weights for combining the selected MSERs at different levels and formed a combined set of MSERs. These combined MSERs provide the initial guess of possible text regions in the input image, In the next phase, we compute three features such as strong edge, stroke-width and edge gradient for individual MSERs corresponding to the initial guess and designed a rule to discard the non-text MSERs of the combined set. The proposed method is naturally scale-insensitive to a reasonable extent. Moreover, it is script independent. Experimental results on the ICDAR 2003 competition dataset have been obtained. Additionally, we simulated the approach on several outdoor scene images captured locally, which contains Bangla and/or Devanagari texts. Finally, we compared the performance of the proposed method with three other state-of-the-art approaches.

Contents

List of Figures

List of Table

CHAPTER 1:Introduction

Text data present in scene images contain useful information. This information may be about the image itself or it can present important characteristics of the ambiance. To automatically extract and analyze the semantic content of the image is another important task in computer human interaction, strongly related to the field of computer vision. The importance of text images can be understood just by following examples. Human beings are able to detect text objects in surrounding environment. Sometimes language may not be known but still the presence of text in an image can be perceived from a decent distance. Scene images containing text can provide people with their geographical location (shop hoarding, maps) , road directions (road signs) , street address, identity of a person (from an i-card), idea about the locality and environment, route details of a vehicle, product details and user instructions (advertisement boards, labels attached to products) and many more. Now manual text extraction by human beings can be hindered by language restrictions as mentioned before. For example, tourist with language barrier in foreign country cannot make out a text in scene images even if it is present. Or for a visually impaired person scene text does not make sense. This is a scenario where automation of text extraction and translation can come to an aid. The constituent steps for automatic text extraction [18] mainly are text detection (to detect if text is present or not in an image) ,text localization (locate the text in the image) and then text extraction from its background and processing segmented text so that it can be fed to OCR (Optical Character Recognition) engine (tool to convert text images to machine readable text). Variety of text font, style, color and sometimes presence of really complex backgrounds in the image make text detection and localization far more challenging and also localised text regions are very noise prone resulting in poor OCR performance. So removal of noise and enhancement of text regions are also well researched topics in computer vision.

1.1 Motivation

With advancement of technology plethora of smart devices have got hold on market. All these devices (smart phones, tablets, smart watch etc.) have revolutionized man-machine interface and have given automation technology all new dimension. These days' smart devices come with extremely sophisticated hardware and software features like multi core processors, razor sharp display, and best quality camera. Thousands of wonderful apps are flooding these smart devices and many of them are examples of image processing applications such as various photo editing apps, OCR engines implemented specially for resource constrained smart phones, video editing apps etc. Automatic text extraction tools can prove to be extremely useful for implementing smart apps. For an example, a person lost in a new place or a tourist with language barrier can take a picture of a street sign using smart-phone and then after locating the text and translating it, can find out where he/she is, through apps like Google maps. Smartphone cameras can be handy in a situation where good quality images are required and can be useful in image processing applications. In the context of this thesis, when scanning of a scene image containing text is not possible, a smart phone can capture the image and this can be used for text localization and extraction modules. In this age of automation and digitization, robotics is also a trending field. The process of making a machine (robot) able to read text from environment is a very challenging task. Auto text processing from scene images are important in case a robot needs to survive in a human environment setting to interpret human language to machine language. So text extraction and retrieving information have become more and more important in modern technology. Moreover the challenges regarding automatic text processing from scene images, like the variety in context, background, ambiance and also font, colour, size makes it a good candidate for intensive research.

1.2 Background

There exist many notable works in the field of text localisation and extraction. Such is the intensity of the research problem that it is in a special category under ICDAR robust reading competition. Eminent researchers have attacked this problem from different directions and dimensions and there have been various famous publications on this topic earlier, some of which are mentioned in this section. Some of the works use connected region based method which itself has two major partition edge based methods and connected component based methods. Another distinct approach is texture based.

One of the earliest approaches in this problem is connected component based approach [13, 22, 23, and 24] since 90's.Connected component approach is interested in finding connected components in the image and then using some text-property non-texts are filtered out. Though edge based method has its own unique approach but still at some stage edge based method also use some connected component detection. However edge based techniques [6, 7, and 10] focus on finding the counter edge as text objects' edges occur in pair. One path breaking innovation in this regard is stroke width transform [6].

Texture based approach [3, 21] takes the image in transformed domain. As text object has different colour and intensity than its background so boundary of such object can be detected using the change in frequency either in wavelet transform [3, 21] or discrete cosine transform [20]. In [20] DCT along with MSER is used to address this problem. Maximally stable extremal region is becoming popular as basic connected component to be detected as text in modern research [9, 20, 25].

1.3 Scope of work

The whole process of automatically extracting text from scene images consists of number of stages. This thesis concerns with the very first and crucial part of the task i.e. to localize and extract the text from the scene images. In the scene images not only texts are present, but also much non textual information are there. Text in scene images appears along with many complex backgrounds. Presence of logo, pictures of human beings or the product or objects pertinent to the textual objects makes the work difficult. For example, a book cover not only contains the name of the book but also some pictures for look and feel or to provide idea about the book content. So with respect to the text region these are noises. To extract the text and translate it the removal of noise is necessary. Some of those noises may possess a regular structure like text. Then it becomes very difficult to separate the noise from the text. But rest of the procedure keenly depends on how much noise-free the data is. The more noise we can remove from the scene image the more pure the image become with respect to text context and the more helpful it is for the next stage of procedure like recognition and translation. This work deals with localization and retrieval of the text and removal of noises as much as possible without harming the text present in the image..

1.4 Objective

This thesis aims to build a method to localize and extract text in scene images. The proposed method is twofold. In the first pass a preliminary mask using MSER (Maximally Stable Extremal Region) detector [1] is built to filter out most of the noise in the scene image. But some MSER detected in this pass contains non textual information that contains textual properties (structures similar to text are not filtered out). Then the image is scanned in various scale space over Gaussian pyramid in search of text candidates. Most consistent candidates are retained and merged to create the mask. Then using that mask we filter the image in the second pass and using some text based property and SWT we remove rest of the non text elements.

1.5 Organization of the thesis

Rest of the thesis is organized as follows –chapter 2 contains literature study, chapter 3 deals with problem formulation, chapter 4 focuses on the proposed approach, chapter5 concern with the experimental results and discussions.

CHAPTER 2:Literature Study

Many good works has appeared, concentrating the problem of text detection, in the literature. Some of them are region based, some are connected component based and some are edge based.

Mao.W et al. **[3]** proposed a method to detect text using local energy variation. They did this on a range of scale. In each scale they wavelet-transformed the image, then found out LL, LH, HL, HH component of images. LL stands for low frequency in horizontal and vertical directions, LH stands for low frequency in horizontal and high in the vertical, HL is for high frequency in horizontal and low in vertical direction and HH stands for high frequency in both of the directions. For each pixel (x,y) local intensity variation and local energy variation is calculated. In LEV analyzed image it was observed that the boundary pixels of objects will have large local energy variations while the pixels in the background or far away from the object boundaries will have small local energy variations. They categorized pixels into three classes i) text boundary pixel having high LEV ii) text like pixel (non-text) having high LEV iii) background pixel. Then thresholding was used to remove pixels with low LEV. Connected component analysis is done to remove all non-text having high LEV using some heuristic rules. Finally multi-scale fusion has been done to get the output.

Zhang and Kasturi **[4]** proposed a method based on character energy and link energy. Character edges are made of pair of edges. Each character is consisting of two set of edges of nearly opposite gradient. Now of pixel in set1 is nearly equal to the cardinality of the other set. Corresponding pixel for an edge pixel of one of the sets can be found from the other set by walking along the gradient direction of the concerning pixel and finding another edge pixel in that way having nearly similar gradient value but in opposite direction. Character energy can be found by two measures; 1) calculating the gradient difference of an edge pixel to its corresponding pixel and sum it up over all pixels then take the average. 2) Counting all the pair who has gradient difference below a threshold and take ratio to the total number of pixels. Using these two measures the formula for character energy is given in equation (9) in the paper. Link energy can be calculated from characters of a string in the image by means of color, size, aspect ratio etc (equation 11 in the paper). Then with this two energies and a threshold text are retrieved from images. But it always assumes that text objects always contain more than one character.

Sivakumara et al. **[5]** distinguished between text and non text by number of strong edges. They divide the video frame in to blocks. For each block they do median filtering and arithmetic mean filtering and found out difference image by subtracting the output of AF from the output of MF. Then number of sobel edge in AF image and number of canny edge in difference image is calculated. For a text block number of sobel edge will be greater. The reason behind this is in text block number of sharp edges is greater. Now even if after AF sobel detector can detect those edges. The difference image has very few canny edges as we have subtracted the AF block from MF. But in case of non text blocks the number of sharp edges is less in AF. So difference between this and the NF block is quite high. Hence the number of canny edges of difference blocks wins over it. However this rule is not strong enough so help of strong edges is needed. A block of weak edges is calculated by subtracting the sobel edges from the canny edges then the number of weak edges is subtracted from canny edges to have the number of strong edges. From median filtering blocks and difference block they found out number of strong edges by canny and sobel detecter. Text blocks will have more strong edges in median filtered block.

Epshtein et al. **[6]** produced the idea of stroke width transform. From each edge pixel they shoot ray toward both positive and negative gradient direction and considered those rays who found another edge pixel of nearly opposite direction. The length of the ray is the probable stroke width of all the pixels along the ray. Now for a text the standard deviation of the stroke widths of pixel will be really small.

Rong Huang et al. **[7]** uses similar idea of ray shooting but they used label histogram of those rays. They labeled the connected components in the edge image. For extracting edge images they used edge preserving smoothing. Canny edge detection is depended on parameter and it may give noises as well. So a bilateral filter is used as EPS filter. Then magnitude image of the filter image is calculated and then canny edge image and binerised version of magnitude image is anded. This filtered edge map is labeled for connected component. Now to join the edges that are not directly connected hey have used Edge quasi-connectivity module. All those labeled edge pair {m,n} are collected at each gap. Keep those which occurs more than a time T_N .now by label union all adjacent label pair are collected together and a new label is assign to them. Then for a CC they shoot ray in both negative and positive directions. For a particular direction label of all the pixels found on other direction of the ray is collected. There label histogram is then calculated. Label histogram of thee rays are a histogram where the in are the labels of the connecting components and the count in the bins are the number of pixel from that labeled component occurring as the end pixel of the ray from the edge pixel of current component. After calculating the histogram four regulation-rules is defined on that. If a connected component passes all those rules then it is declared to be a text component.

Cong Yao et al. **[8]** gave an approach to detect text in any orientation. They used SWT in initial stage to find candidates. The next stage of component analysis was of two phase. In 1st phase depending on the height, width of the bounding box of CC and aspect ratio, occupation ratio non-texts are rejected. 2nd phase consisted of a trained classifier capable of identifying text object and rejecting non-texts objects. The features were rotation invariant, scale invariant and computationally of low cost. To have such features estimation of the center, characteristic scale and major orientation of each components are calculated. For a component it barycenter, major axis, orientation are estimated using Camshift algorithm[16]. Using those measures, features are calculated. These component level features can be found in the paper at section3.2.1.Then candidates are linked in to pairs. A greedy agglomerative clustering was used for this purpose. For each pair of components their orientation consistency, population consistency is found as per the equation (1, 2) in the paper. Similarity of the components can be found as the convex combination of these two characteristics. According to this similarity definition chains with proximal sizes and orientations are merged. To get more accurate result at chain analysis phase a classifier trained with chain level features is used. Total probability of each chain is then calculated based on the probability by initial classifier result and result of classifier in this stage. The chains whose probability is lower than a threshold is discarded.

C.Shi et al. **[9]** combines the idea of MSER and graph-cut algorithm. They first detect the MSERs. Then build a graph with those MSERs as nodes and two other terminal nodes stands for background and foreground. The edges of the graph are defined by means of various cost function which incorporates region based as well as context-relevant information. Cost functions are majorly divided into two categories i) unary cost function ii) pair wise cost function. Now MSER labeling problem can be thought of as segmenting problem by labeling text as 1 and non-texts as 0. Given the graph whose nodes are the MSERs as well as the two terminals, the cost of labeling each node as foreground or background could be calculated and it could be minimized by finding the minimum cut of the graph. Now minimum cut of the graph gave the segregation of text and non-text MSER. Then text grouping is done based on the similarity of color, height, maximum and minimum column index and rows of areas' centers of each component pair. A trained classifier is used at the final step, to remove the noises which are not being removed by previous all this steps, for better performance.

Chunmei Liu et al. **[10]** used an unsupervised edge based technique. They extract the edge information in four directions (0, 45, 90, and 135). They compute 24 features (6 from each edge image) and use K-Means clustering with 2 clusters to localize the text.

Sivakumara et al. **[11]** uses Fourier-Laplacian filter to extract high frequency information near the transition of text to background and background to text. They transformed the image in Fourier domain then discard noises by filtering with a ideal low pass filter. However this is equivalent to Laplacian of Gaussian. Them they found maximum difference [14] using a sliding window of a fixed size. The size of the sliding window is the approximate stroke width of which text we are interested to detect. And then k-means clustering with cluster number 2 gave the initial segmentation. Then depending on the number of junction pixel or intersection pixel on skeleton image of a connected component, connected components were classified into two class; simple and complex. According to type of CC necessary measures has been taken to omit the false positive and get the final output.

J. Zhang and R. Kasturi **[12]** uses edge gradients of CC edge images to eliminated noises in the first pass. They group all possible gradients of edge pixels in four groups. They put some heuristic rules on those four measures with respect to some thresholds. In next step they built graph with CC as nodes and similarity between two CC as the edge of the graph. The more similar they are the more weight the edge possesses. Then from eigen value and eigen vector (positive) of the weight matrix of the graph they draw their decision of text-non-text classification.

Roy Chowdhuri et al. **[13]** presented a scheme based on distance transformed. They tried to estimate stroke width using distance transformed image. They first extract the edge image of the input image. Then they used morphological closing for joining small discontinuities in the edges. Then for each CC sub image they find the distance transformed image of which background is darker than foreground. For each pixel in DT they found if that is the local maxima or not. If so then the DT value is stored. Now if the mean of newly stored DT values is greater than the twice of its standard deviation then the sub image is declared as text with further processing on some textual and geometric property.

Renwo Gao et al. **[19]** proposed a method based on saliency map. This method is an application of the Model of Saliency-Based Visual Attention, proposed by Itti et.al [20]. It uses saliency map in two level. In first step a saliency map is calculated from the input image. then region of interest is evaluated and all pixel s are automatically classified in to two class 1- for pixel in saliency map 0 for other to build a mask. Using this mask image is filtered. This filtered image is input to the second step. In second step again a saliency map is obtained using the filtered image.

Wang et al. **[20]** proposed an approach which incorporate MSER component along with HOG features. The whole procedure is consisted of four steps. Component extraction, the 1st steps, uses MSER detectors to get the basic connected components from the image. Component Dictionery classifier is trained based on HOG feature. For training the classifier samples are clustered in K classes in HOG feature space using K-means clustering. A multi-class SVM training algorithm is used to train the dictionary classifier that contains K linear SVM that corresponds to K component clusters. Component consensus feature are then extracted. These are pair wise relation among the components and holistic variance of grouped components. An integrated discriminative model is built based on the classifier and the consensus features. An algorithm to detect text based on this model is proposed finally in Section E of the paper.

CHAPTER 3:Problem formulation

The problem is formulated in two major phases.

1. Mask Generation.
2. Text Extraction.

3.1 Mask Generation

In mask generation phase initial mask was generated. This mask consists of text candidates and some noises. We tried to remove the noises as much as possible in this phase. The mask build in this phase will be use in the next phase for further extraction of the text. This phase majorly deals with *Gaussian-level* decomposition and the *MSER* detection in each level of the Gaussian pyramid. Before entering into the decomposition we resize the image. The block diagram of the mask generation phase is given in Figure 1.

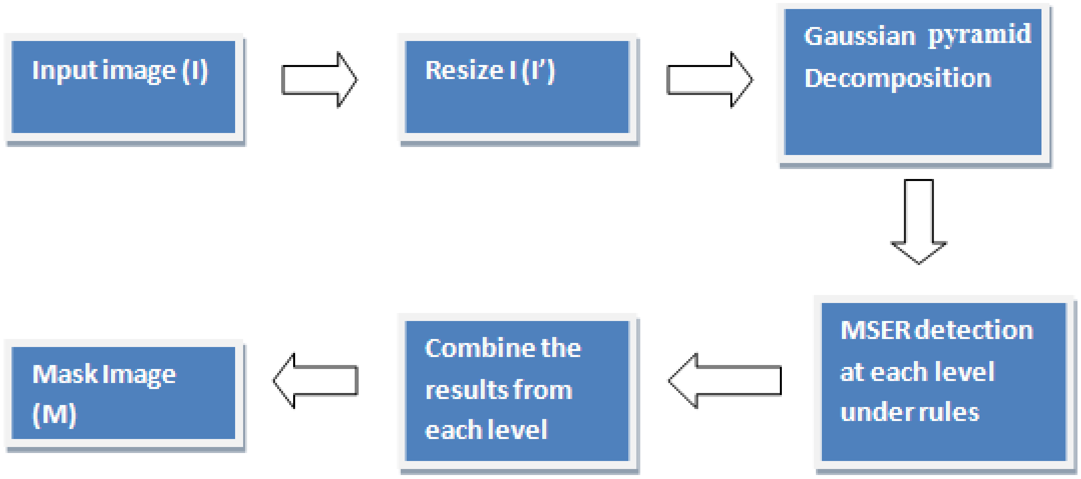

Input image (I)
Resize I (I')
Gaussian pyramid Decomposition
MSER detection at each level under rules
Combine the results from each level
Mask Image (M)

Figure 1: Block diagram of mask generation

3.2 Text Extraction

In this phase the mask is used to extract the likely text MSER from the text. Now those texts MSER are further scanned under a set of rules to be sure about whether they are text or non-text. This phase consist of mainly 4 modules as follows,

1. Edge filtering.
2. Stroke width consistency checking.
3. Histogram of oriented gradient checking.
4. Strong edge criteria checking.

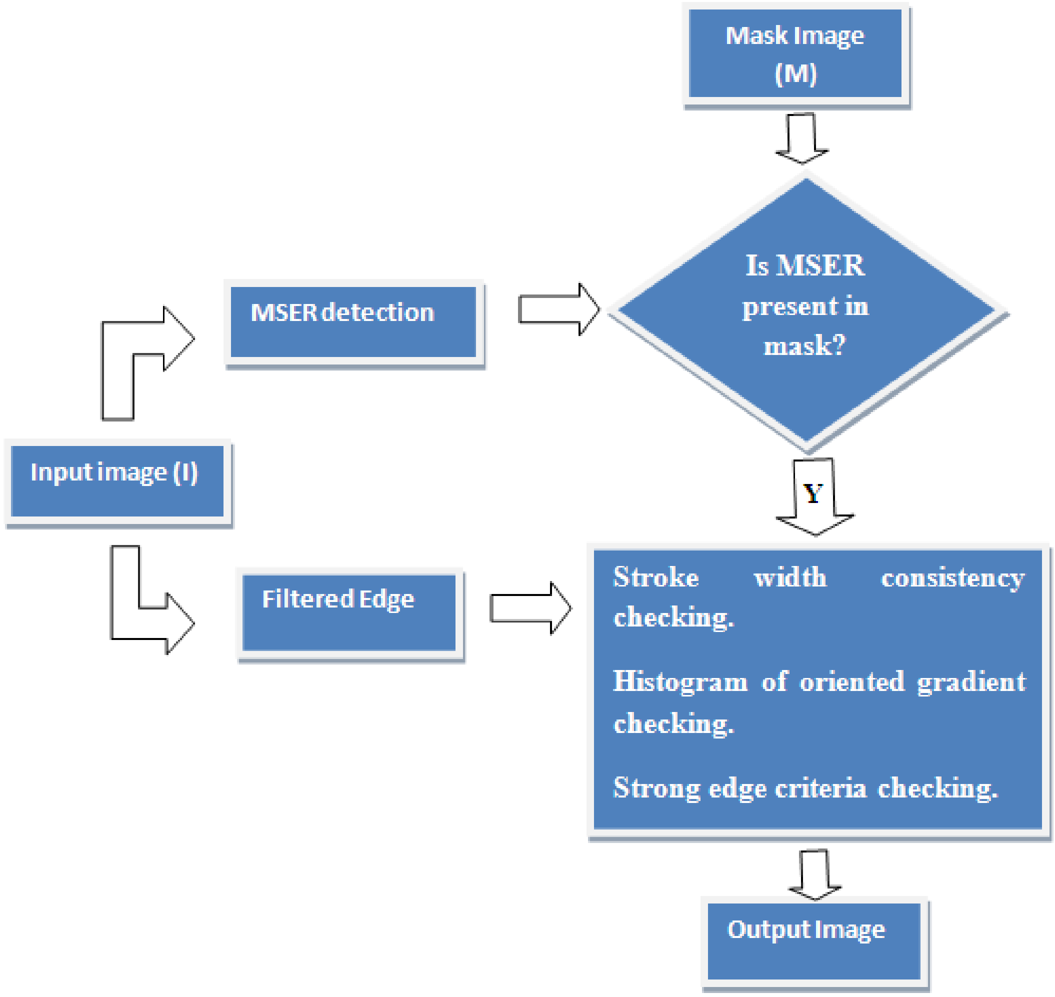

Mask Image (M)
MSER detection
Is MSER present in mask?
Input image (I)
Y
Filtered Edge
Stroke width consistency checking.
Histogram of oriented gradient checking.
Strong edge criteria checking.
Output Image

Figure 2: Block diagram of text extraction phase

CHAPTER 4:Proposed Approach

As we have discussed in previous chapter, the proposed method consists of two phases, the organization of this chapter will be in that sense only. The method tries to find out the MSERs which persist over some level of MSER fulfilling some properties. Before entering into the details of the procedure a small introduction of Gaussian pyramid decomposition and MSER is necessary.

Gaussian pyramid decomposition is a widely used image decomposition technique used in image processing domain. At each level the image is a “reduced” version of previous level’s image, reduced in the sense both resolution and sample density is decreased. The very beginning level of the pyramid contains the actual image. Each value within a level is computed as weighted average of values in previous level within a 5×5 window and then sub-sampled. Detailed about Gaussian pyramid [2] can be found in appendix A.

Extremal regions of an image can be defined informally as that region in image which has either greater intensity or lower intensity than all the pixels on its boundary. Now if we binerise such a region with respect to a suitable threshold then depending on whether the extremal region has higher or lower value than its boundary, it will be assigned 0 or 1 value. Now assume in the image all extremal regions have higher value than its boundary. Now starting with a very low threshold, thresholding of the image will give only a single region. Now if threshold is increased gradually then region will start to break up. Now if a tree is built [17], in which a level is the value of thresholding and nodes are the disjoint regions found in that thresholding with parent child relation-‘a child of a region A in level l is B_1, B_2, B_3, …, B_n in l+1 such that $B_1 \cup B_2 \cup B_3 \cup \ldots \cup B_n$ is a subset of A and thresholding value increases as we move down along the tree. A maximal stable extremal region is that one which has minimal value for a stability function along the path of the tree[17]. Mathematical definition of MSER can be found in appendix B.

Now with this preliminary knowledge of the basic component we can delve into details of proposed method.

4.1 Resize the image

Resize of the image is done just before it is sent for Gaussian decomposition. As discussed earlier at each level of Gaussian pyramid image dimension get reduced by a factor of 2 as a result of sub-sampling. So to keep ease over the iteration through the decomposition procedure image dimension is made power of 2 and image is made of square dimension. The color image is first converted into gray level image. Then this resizing of dimension take place. The appropriate power of 2 to which the image will be resized can be found out by the given formula-

Find an m such that mЄ$\mathscr{J}^{+}$ and | 2^{m}-height|+|2^{m} –width| is minimum, where $\mathscr{J}^{+}$ is the set of positive integers, height is the number of rows in the image; width is the number of column of an image if image is considered to be a two dimensional array.

4.2 A detailed view over mask generation

Text in images has distinct contrast than its background and more or less of uniform intensity. Hence they are likely to be MSER in the image. So MSER detector is a natural choice to detect text in images. But not only text images appears as MSER but also some noises or cluttered element from background may appear as MSER. So to get rid of those we introduce the concept of stable MSER over scale space.

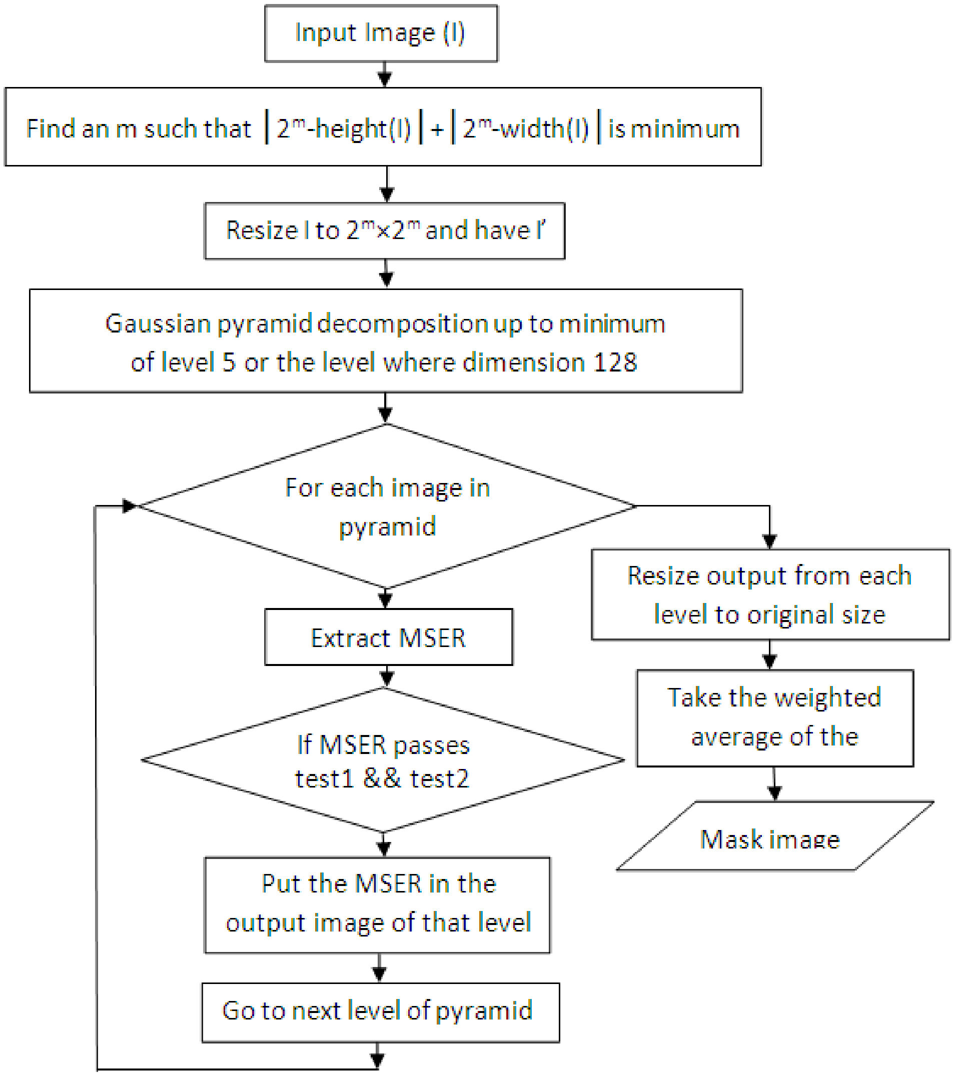
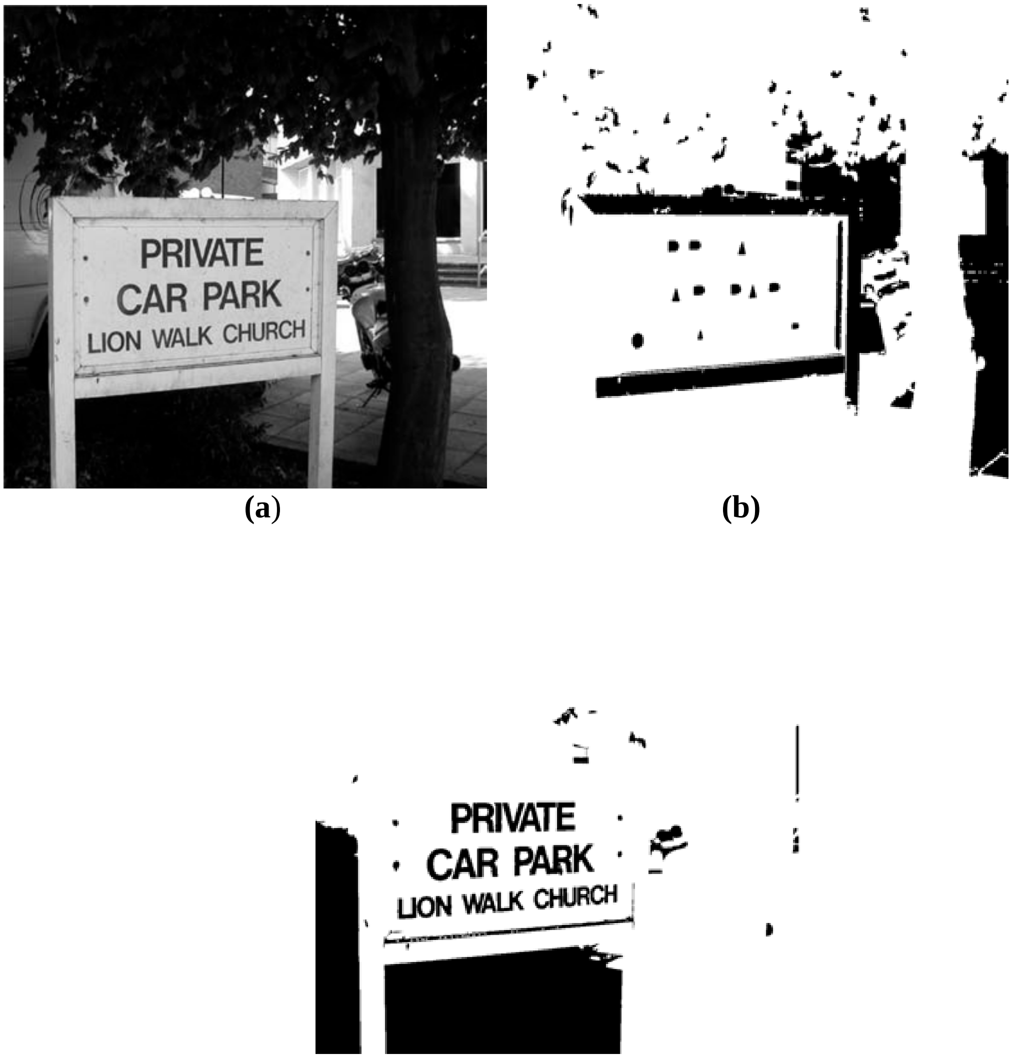

Input Image (I)
Find an m such that |2^m-height(I)| + |2^m-width(I)| is minimum
Resize I to 2^m×2^m and have I'
Gaussian pyramid decomposition up to minimum of level 5 or the level where dimension 128
For each image in pyramid
Extract MSER
If MSER passes test1 && test2
Put the MSER in the output image of that level
Go to next level of pyramid
Resize output from each level to original size
Take the weighted average of the
Mask image

Figure 3: Flow chart of Mask Generation



(a)

(b)

(c)

Figure 4: Images from 1st level of pyramid. (a) Gray valued image (b) bright MSER s (c) dark MSERs

4.2.1 Decomposition of image and MSER extraction

We decompose the image up to that level of pyramid where the dimension become128×128. In each level of pyramid we detect bright MSER and dark MSER. MSERs with too big size are not considered for further use. Each of the MSER of either type is then scanned under Test1 and Test 2. Those MSERs which pass both the test are kept in a separate image as output of that particular level. Towards the end of the process output images from each level is combined to make the mask image.

Figure 4, 5, 6 give a view of output over the levels of Gaussian pyramid. Note the dimensions are rescaled approximately to fit in this page. The motive is to give the basic insight view of MSER images over the levels of Gaussian pyramid. In these Figures the MSER images contains all MSERs of that particular type (either dark or bright). No tests or rules have been imposed yet.

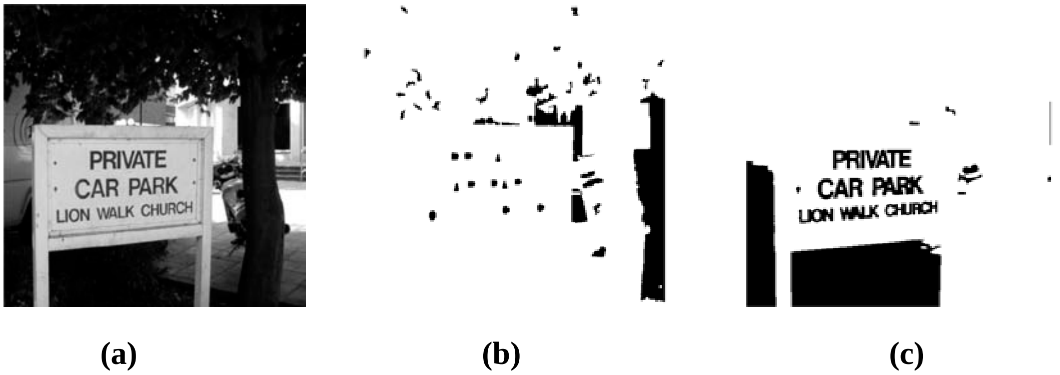
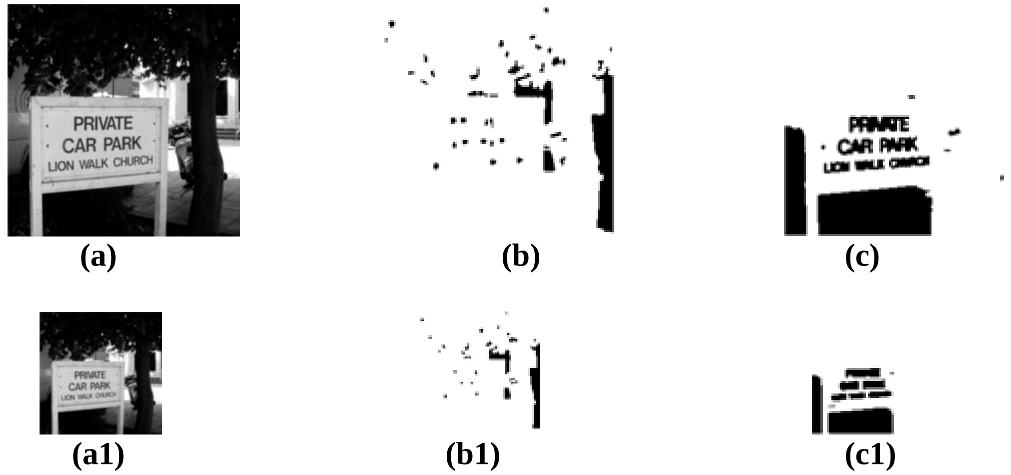


Figure 5: Images from the 2nd level of pyramid (a) gray value image (b) Bright MSERS (c) Dark MSERs

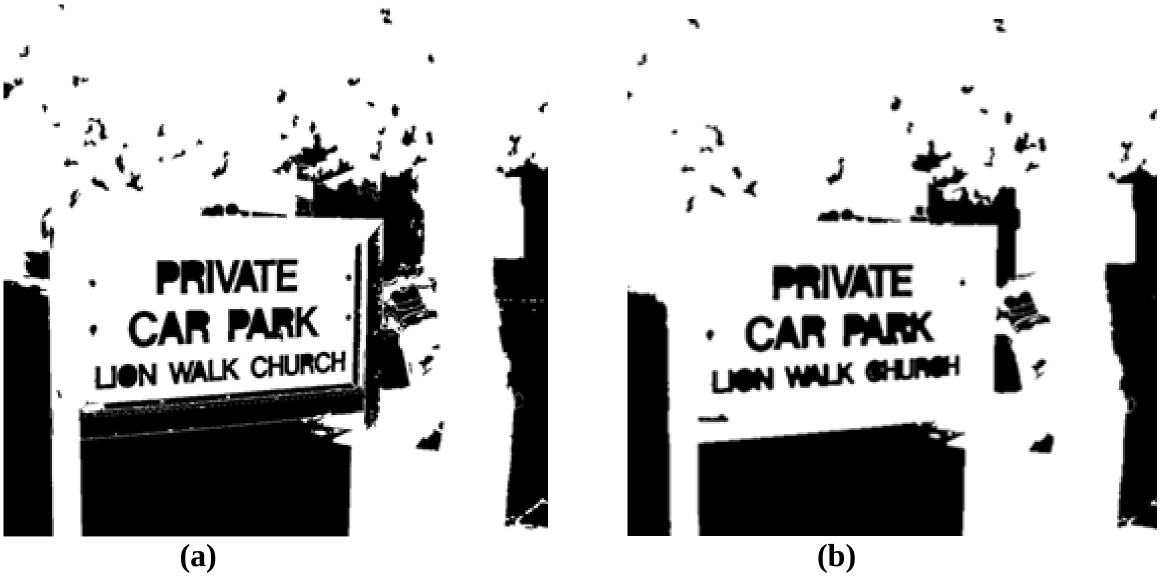


(a) (b) (c)

(a1) (b1) (c1)

Figure 6: Images from 3rd and 4th level of Gaussian pyramid

Figure 7 depicts the difference between combined image obtained from the each level from the MSER and if MSERs from actual image had been used as a mask. At each level dark MSERs and Bright MSERs are put together to make the output of that very level. Now output from each level is combined to give that combined mask. Details about this can be found in *section 4.2.4*. Note that none of them yet go through any tests or rule but still in combined image noise is less.



(a) (b)

Figure 7: (a) image of all MSER from actual image (b) image of combine MSERs from pyramid levels

Advantage of using pyramid

One thing is depicted clearly from Figure 7 that via use of MSERs over the pyramid levels helps to remove noises. But this is not the only advantages coming from the approach. Sometime it helps to retrieve lost information too. At the end of section 4.2.1 the information regarding the second advantage is furnished.

4.2.2 Test1

This is for checking whether the MSER can be a valid text or of any arbitrary shapes. Here decision is taken on the basis of three measures as discussed below.

Occupation Ratio

It is ratio of the concerned MSER pixels to the total area of the bounding box of the MSER. Text MSER occupies a decent amount of space in its bounding box. So MSER with too small occupation ratio or too big occupation ratio could not be text. We put lower limit as 0.3 and upper limit as 0.8 for occupation ratio.

Our next two features, intended to capture "*Regularity*" of text candidates, are inspired from [9] equation 5, 6 with some little changes.

Fea(1)

It is ratio of number of pixels present in the skeleton of the MSER to that of edge pixels (instead of contour pixels) of the MSER. As mentioned in [9], text should not have too random or complex structures; this ratio is expected to have greater value for text MSER. In our process we put a threshold beyond which an MSER having value for this feature is considered.

Fea(2)

It is ratio of the area under the bounding box of the concerned MSER to the square of number of edge pixels. However, we find it to be lower than a threshold value for text MSERs experimentally. As the threshold is fixed but through pyramid level dimensions and shape and size of MSER changes, some text MSERs even can go beyond that threshold. So we divide this ratio with 2^L, where L is the current level of pyramid. So at very initial stage that is at level 0 we are considering the actual ratio, and then it is get divided by 2 at each level. An MSER is selected if it follows all the three criteria .Flow chart of test 1 is given in Figure below

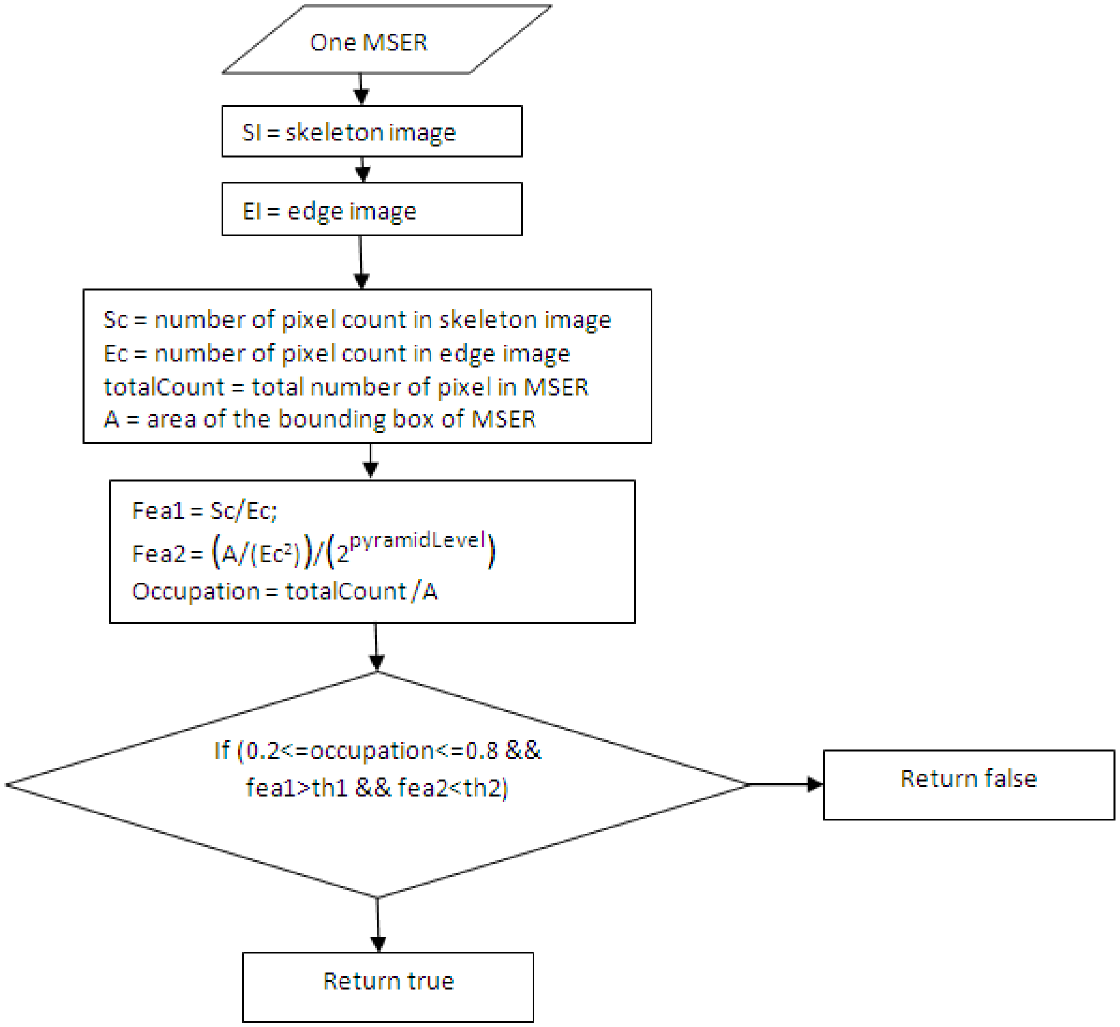
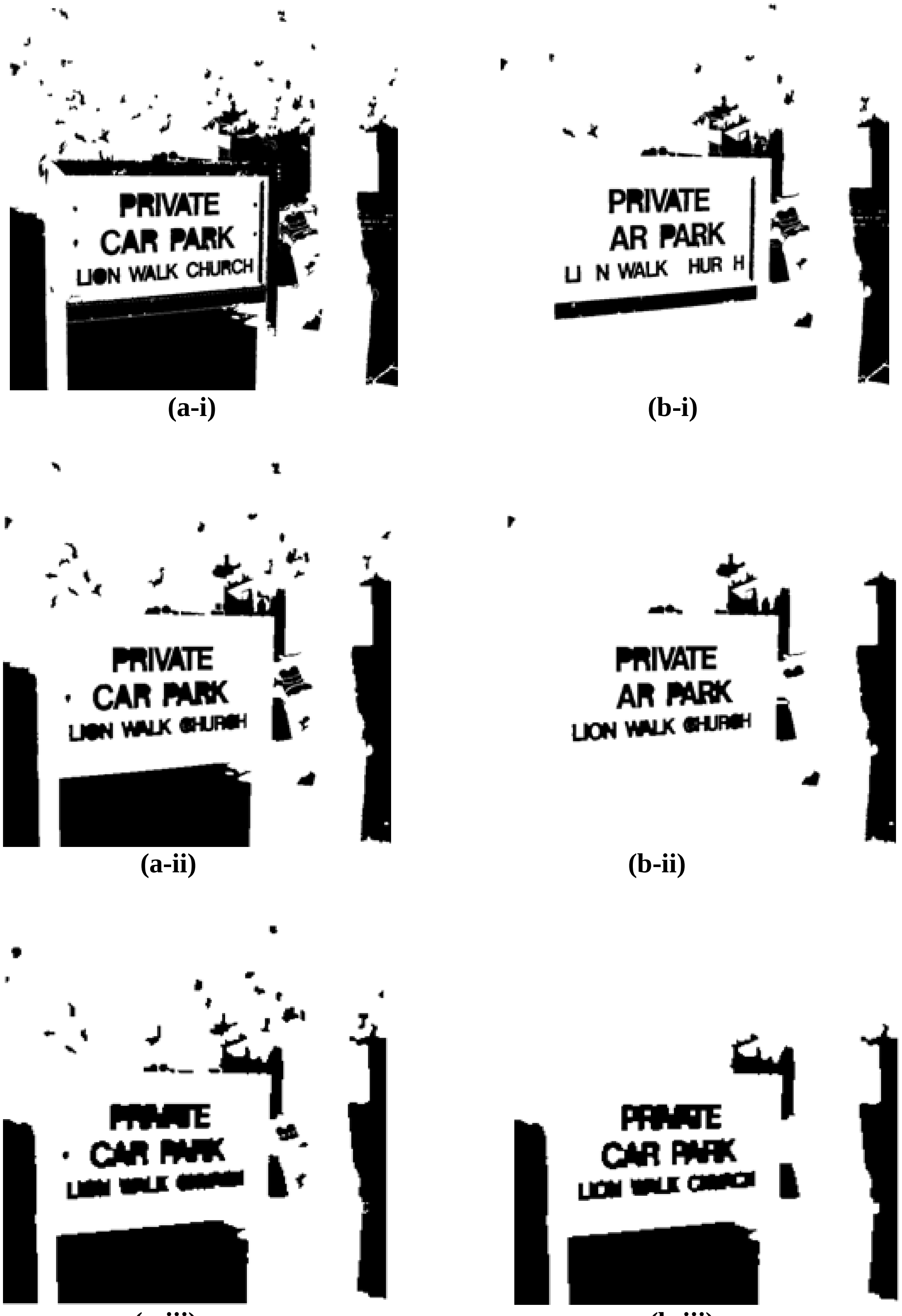

One MSER
SI = skeleton image
EI = edge image
Sc = number of pixel count in skeleton image
Ec = number of pixel count in edge image
totalCount = total number of pixel in MSER
A = area of the bounding box of MSER
Fea1 = Sc/Ec;
Fea2 = $(A/(Ec^2))/(2^{pyramidLevel})$
Occupation = totalCount / A
If (0.2<=occupation<=0.8 && fea1>th1 && fea2<th2)
Return false
Return true

Figure 8: Flowchart of test1



(a-i)

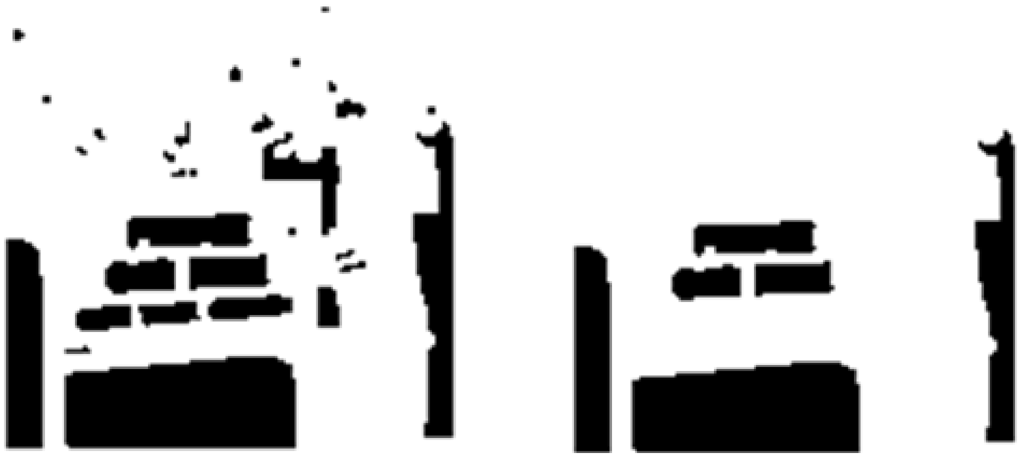
(b-i)

(a-ii)

(b-ii)

(a-iii)

(b-iii)



(a-iv) (b-iv)

Figure 9: a) Output from pyramid levels before test 1 b) Outputs from pyramid levels after

Advantage of using pyramid

Figure 9 is strong enough to enlighten the reason of adopting MSERs over pyramid levels rather than directly on the actual image. Note at each level using test1 noises has been reduced. But also some data has been lost which is a negative impact (note b-i and b-ii output from the 1st two level of pyramid after test1). But the lost data are retrieved from the next two levels. Similarly there are some data getting deleted in last two levels but in final image they come as contribution of images from other level. If single level of MSER would have been used lost data cannot be retrieved. And then data has to be retained at the cost of allowing noises.

Though the reduction of noise is evident from the Figure 10 but a shape distortion of some object in the image cannot be avoided by bare eyes. The logic behind this can be explained with help of the Figure 9. Note that the character ‘C’ present in the all the four level before introducing test1 (see Figure 9-a(i-iv)). As per the combination rule at final output the shape of the ‘C’ dominated by level 1 and 2. But from Figure 9-b (i-iv) it can be noticed that ‘C’ is retrieved from level3 and 4. So the shape is dominated and approximated from those levels only. That is why there is a shape distortion in ‘C’ and many more. However the prime aim of this stage is to make a mask which is satisfied till now.

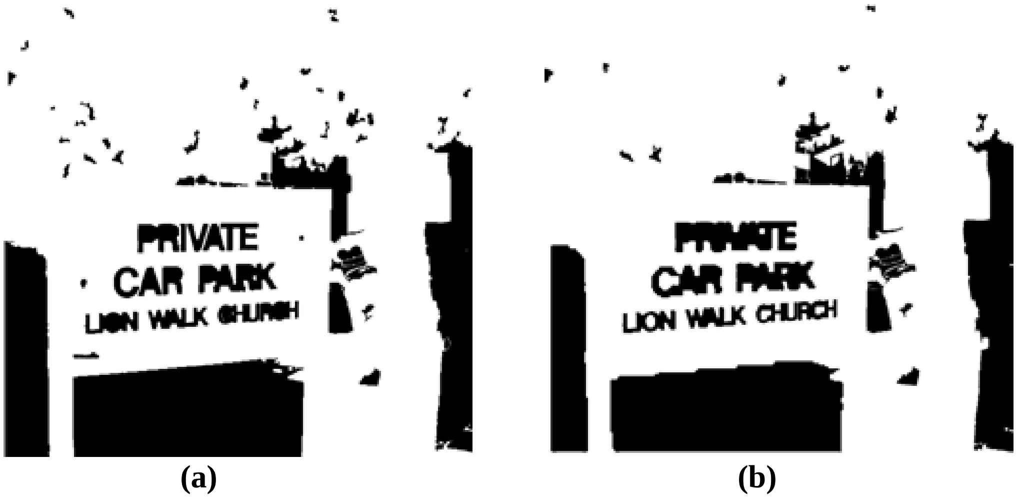
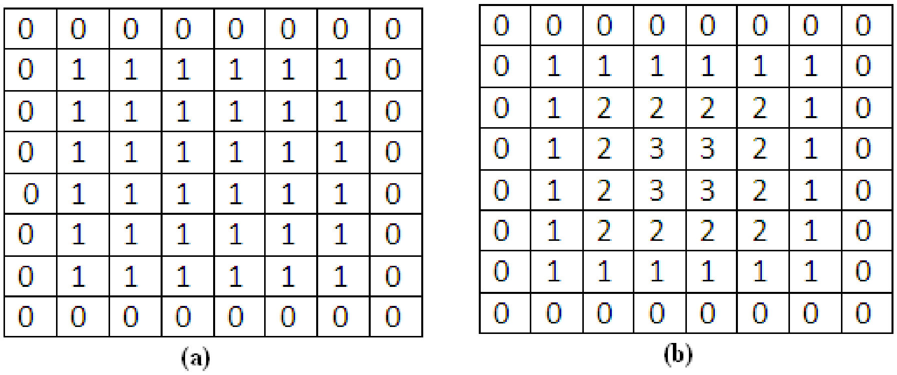


Figure10: (a) Combined mask before test1 (b) Combined mask after test1

4.2.3 Test2

This test discriminates between text and non-text MSERs using the idea used in [13]. Each MSER is subjected to the Euclidean distance transform (DT) [15]. Each pixel in the resulting image is set to a value equal to its distance from the nearest background pixel. An idea about this can be found in the image below. In this Figure 11-(a) is a simulated sub-image having value 1 for object pixel and 0 for background pixel. We compute the distance of each object pixel from its edge or boundary as shown in Figure 11-(b).



0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0

(a)

0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0
0	1	2	2	2	2	1	0
0	1	2	3	3	2	1	0
0	1	2	3	3	2	1	0
0	1	2	2	2	2	1	0
0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0

(b)

Figure 11:(a) sample of an image with foreground1 and back ground 0. (b) DT transform image of the image in (a)

However, unlike the process mentioned in [13] we do not need to binerised the CC (in our case MSERs) as they are already thresholded image. As out MSER pixels are always white and background is black we do not need to judge between DT image of complement of the MSER sub-image and the actual MSER sub-image.

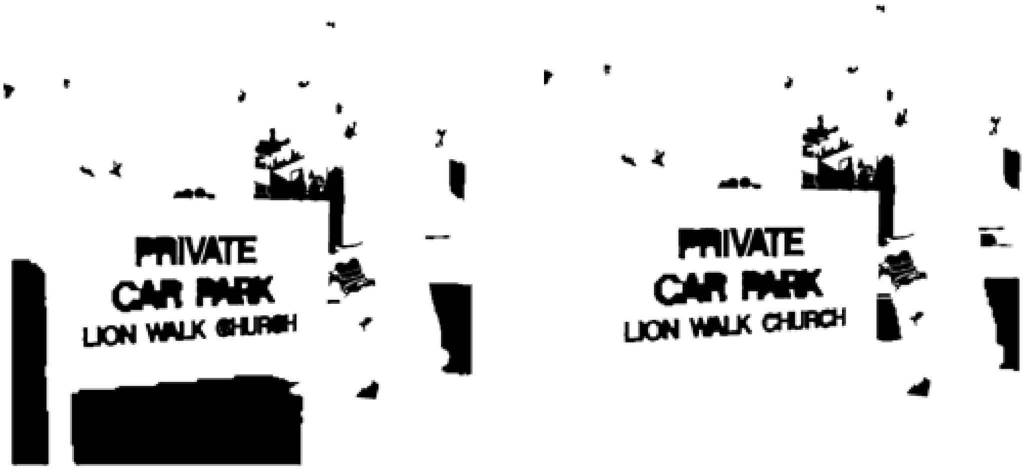
As discussed in [13] we also consider a 3×3 window around each pixel in DT image. In [13] DT value of a pixel is collected if it is the local maxima within the 3×3. But in our case we consider the value of the local maxima irrespective of the fact whether the center pixel possesses that value or not. This makes the occurrence of local maxima more. The advantage of collecting the local maxima sis depicted in the Figure 8 where (a) is proposed by [13] and (b) by procedure of collecting local maxima of ours. Now we collect all such value in a ser <T>. And calculate the mean and standard deviation. In [13] if mean is greater than twice of standard deviation then the CC is declared to be text. As said earlier in our case dimensions of MSERs are changing from level to level of the pyramid, we consider mean to be greater than a ratio ;

MSER = text; if mean ≥ ratio * std;

= Non-text; otherwise

Ratio = max [h/w, w/h]

h is the height and w is the width of the bounding box of the MSER. Let say this to be rule 1.



(a) (b)

Figure 12 : a) collecting DT value if it is local maxima in 3×3 window in DT image. b) Collecting the local maxima in 3×3 window in DT image.

Rule1 alone, however, not strong enough to remove all the noises. So we collected all those pixels which have same distance from background either in left -right direction or in up-down direction or in any diagonal direction. In a word we are collection those pixel which are approximately middle of the object in each row. So we are approximating the skeleton of the MSER. Now DT values of those approximate middle pixel are collected in a set <T'>. Mode of the set <T'>. Now if the size of <T'> is greater than a threshold

(a_1) with respect to minimum of h and w, and if mode is within the 2-sigma limit of mean then we consider the value of the mode as probable stroke width. Otherwise no such thing exists. The logic of comparing the size of <T'> to height and width is for a text candidate number of skeleton pixel is sufficient with respect to its dimension. Moreover in case of a text character as they are of uniform stroke width so there will be good number of those middle most points. However for non-text cluttered shape number of such pixel is very less as distribution of background pixel is not quite uniform. Now if the probable stroke width exists and if it is less than a_2×min(h, w) then the MSER is selected. The flow chart of test 2 is given below.

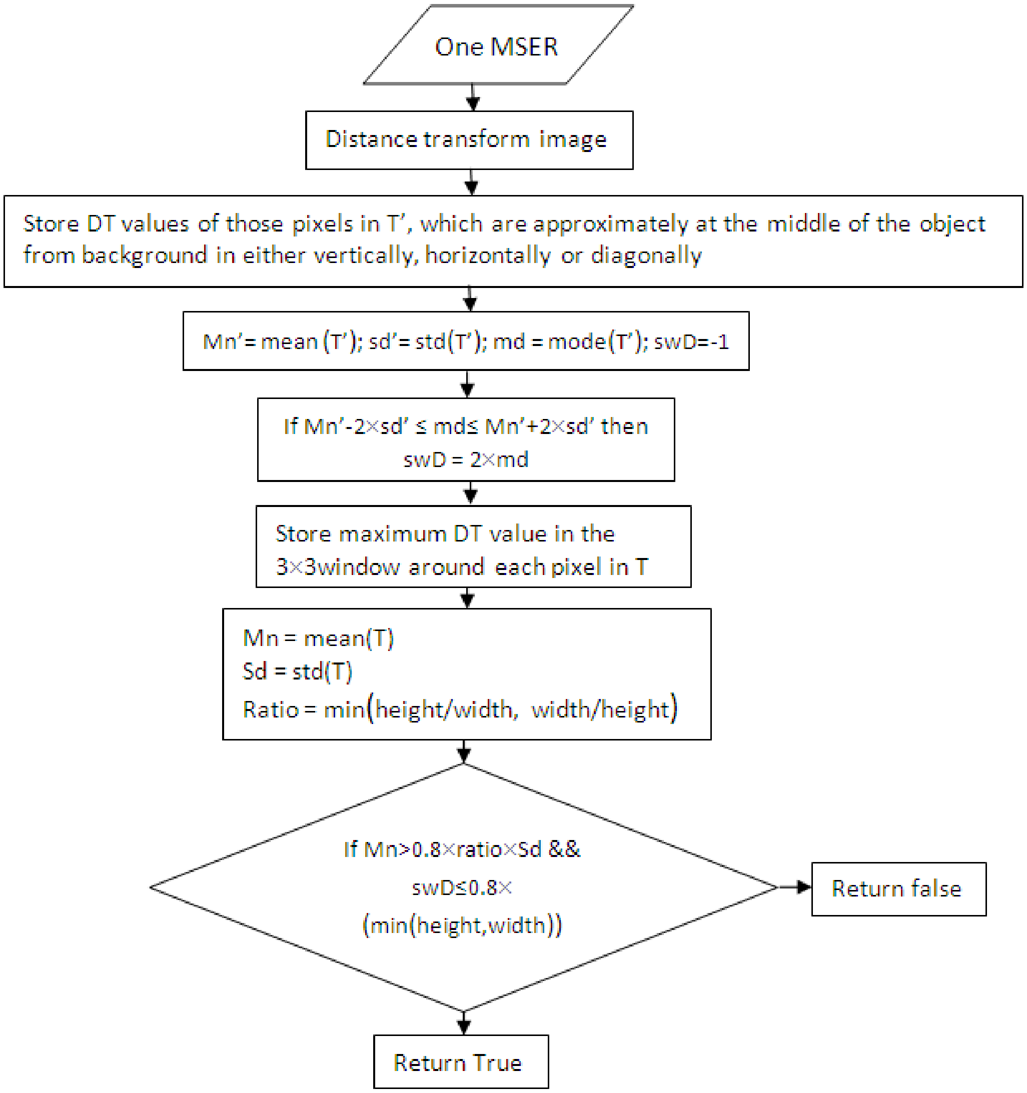
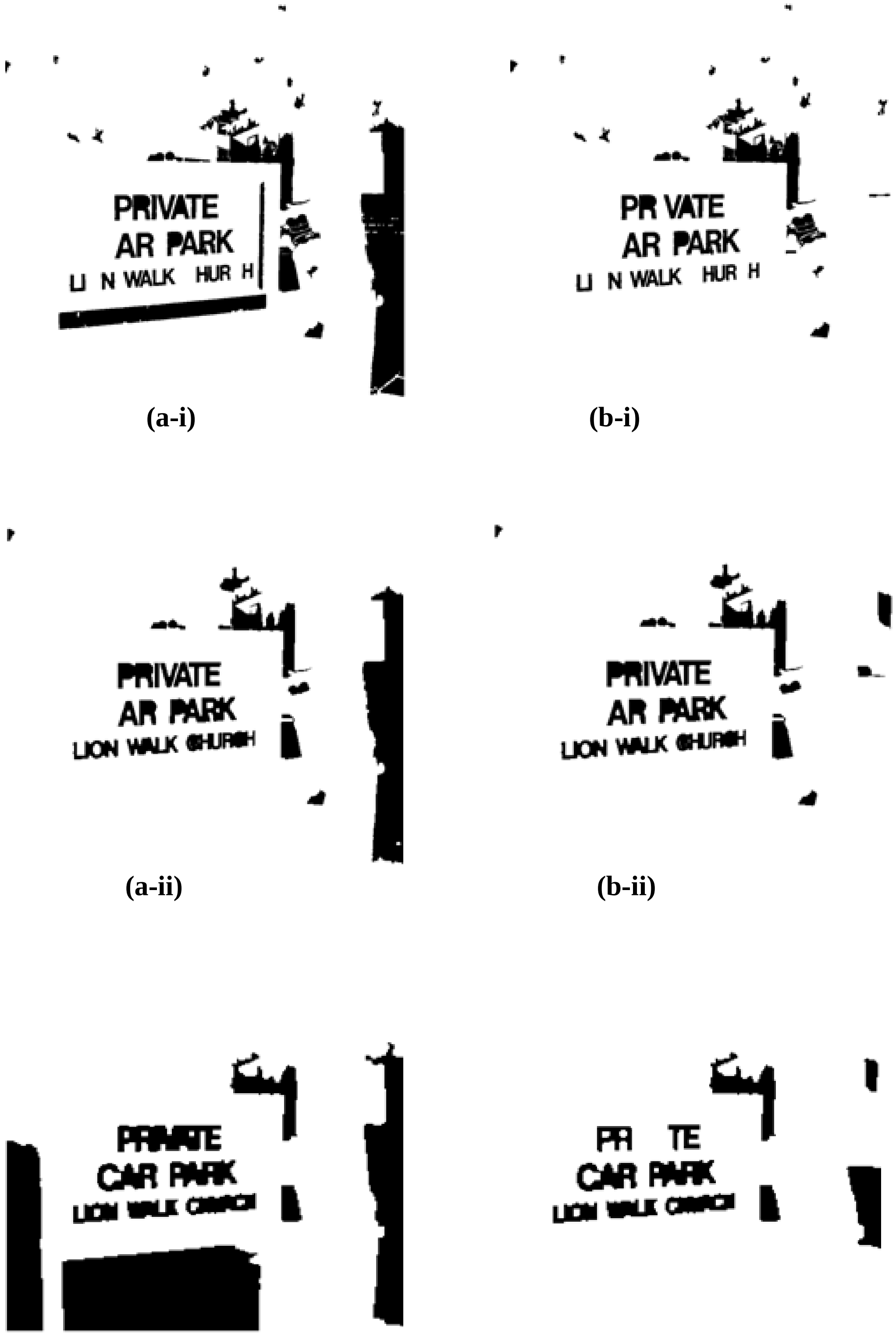

One MSER
Distance transform image
Store DT values of those pixels in T', which are approximately at the middle of the object from background in either vertically, horizontally or diagonally
Mn'= mean (T'); sd'= std(T'); md = mode(T'); swD=-1
If Mn'-2×sd' ≤ md≤ Mn'+2×sd' then swD = 2×md
Store maximum DT value in the 3×3window around each pixel in T
Mn = mean(T)
Sd = std(T)
Ratio = min(height/width, width/height)
If Mn>0.8×ratio×Sd && swD≤0.8× (min(height,width))
Return false
Return True

Figure 13: Flow chart of test2



(a-i) (b-i)

(a-ii) (b-ii)

(a-iii) (b-iii)

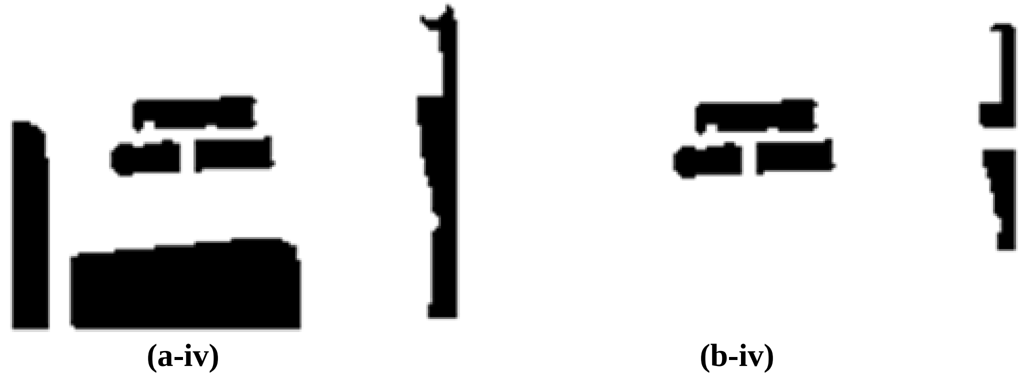


Figure 14: a) Output from pyramid levels after test 1 and before test2 b) Outputs from pyramid levels after test2

If rule1 and rule2 is satisfied then MSER passed this test2.

4.2.4 Combining the result

An MSER is selected if it promoted by both test1 and test2. At each level of pyramid whole process is done for each eligible MSER (above a certain size) from both kind of MSERs (dark on bright background and vice versa) and then all the MSER (bright and dark) are merged to form a single mask image at that particular pyramid level.

The iteration over pyramid levels are continued until the minimum of its dimensions reached to 128. Because below that dimension either the whole image become a single MSER or there may be no MSER in the image. So combining a mask of all white may select some noises, while a mask of all black may omit some data during combining the results. In both sense the mask become useless. Now we need some way to combine all the masks over the pyramid level. So before combining mask of any level it is resized to original image dimension. Initially we just sum them all and took just the average to produce ultimate mask.

But in our practical life, the nearer we are to the scene the more ability to read a text. So inspired by this nature, we put more weight to the masks at lower level. The weighted average scheme of ours is as follows.

```
Weight = 1;
sumWeight = 0;
for $i$ = 0 to $L$
    w = Weight/$C$;
    imageFinal = imageFinal + w × Mask($i$);
    Weight = Weight – w;
    sumWeight = sumWeight + w;
end
imageFinal = imageFinal/ sumWeight;
```

Where L is the number of level of pyramid up to which minimum of image height and width remains greater than equal to 128. Mask(*i*) denote the mask image obtained at pyramid level *i*. *C* is a constant. Too lower or too high value for C is unsuitable. We have done our experiment on C = 5.

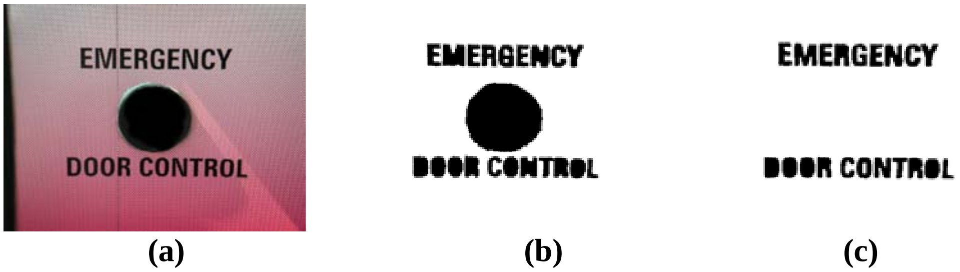


Figure 15: Demonstration of different value of C. (a) original image (b) mask with C=3 (c) mask with C=5

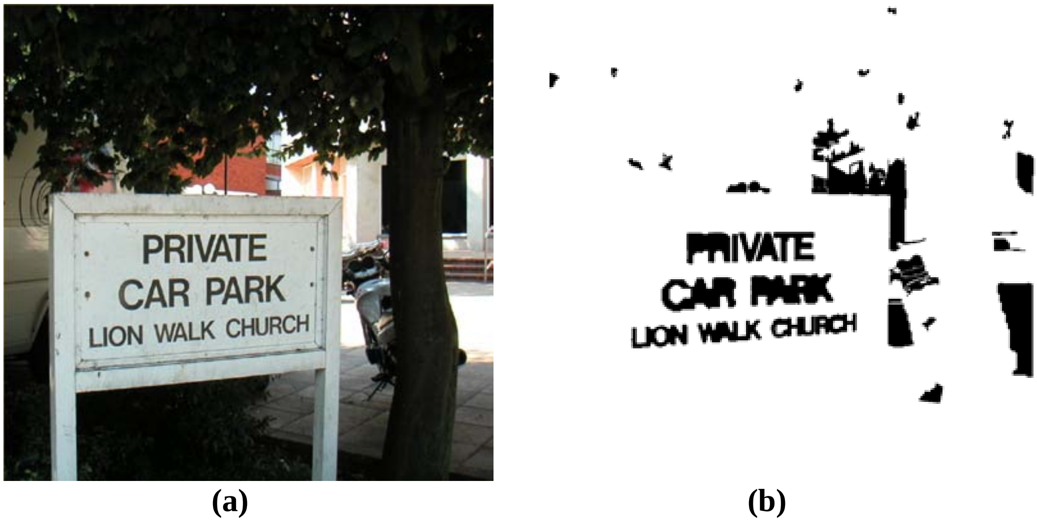


Figure 16: (a) Original image (b) Mask created by proposed method

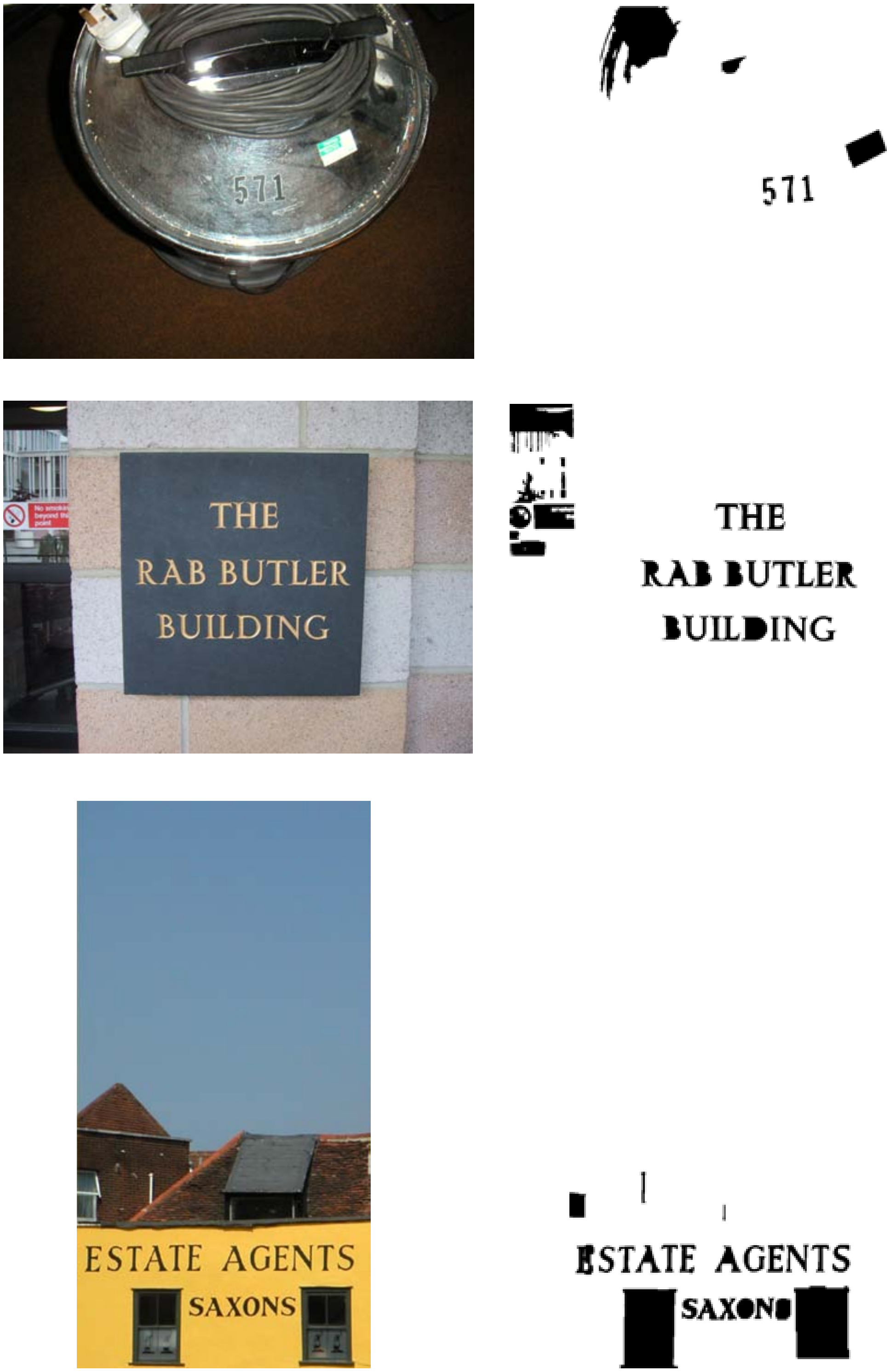
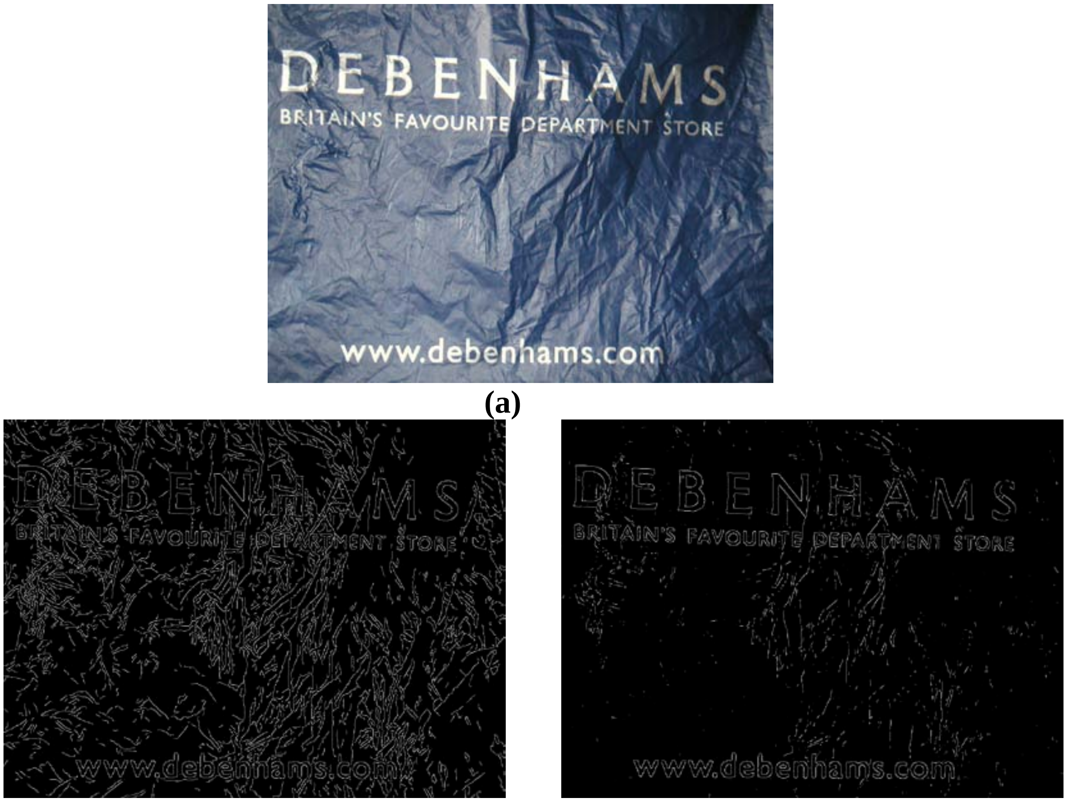


Figure 17: Examples of some masks given by proposed method

4.3 Text Extraction

Up till now very few of textual properties have been used. So noises in the final mask are not something out of blue. So to get rid of these unwanted things in final images some properties of the text are needed. In this context we used the idea of HOG based text extraction from [12], stroke width [6] and the concept of strong edge [5]. The detail of the process is as following-



(a)

(b) (c)

Figure 18: a) Original image b) Canny edge image c) Image after edge filtering

4.3.1 Edge Filter

For HOG based text extraction and stroke width estimation needs single width edge pixels of those regions corresponding to selected MSER masks. Canny edge detection depends on two parameters. And it may sometime be noisy. So to get out of this problem we use the edge detector described in [7]. We use bilateral filter to have edge preserving smoothing. Now Otsu's thresholding is applied on the magnitude image of this bilateral-filtered image. This binerised image is then anded with output of canny edge detector with low thresholds. The effect of the filtering is depicted in Figure 18.

4.3.2 Final MSERs

From the gray image of the original image MSERs are once again detected. However output from the 1st level of Gaussian pyramid can also be used as a substitute. Now those MSERs which are present in this current image and as well as in the mask that has generated in previous section are used for further studies.

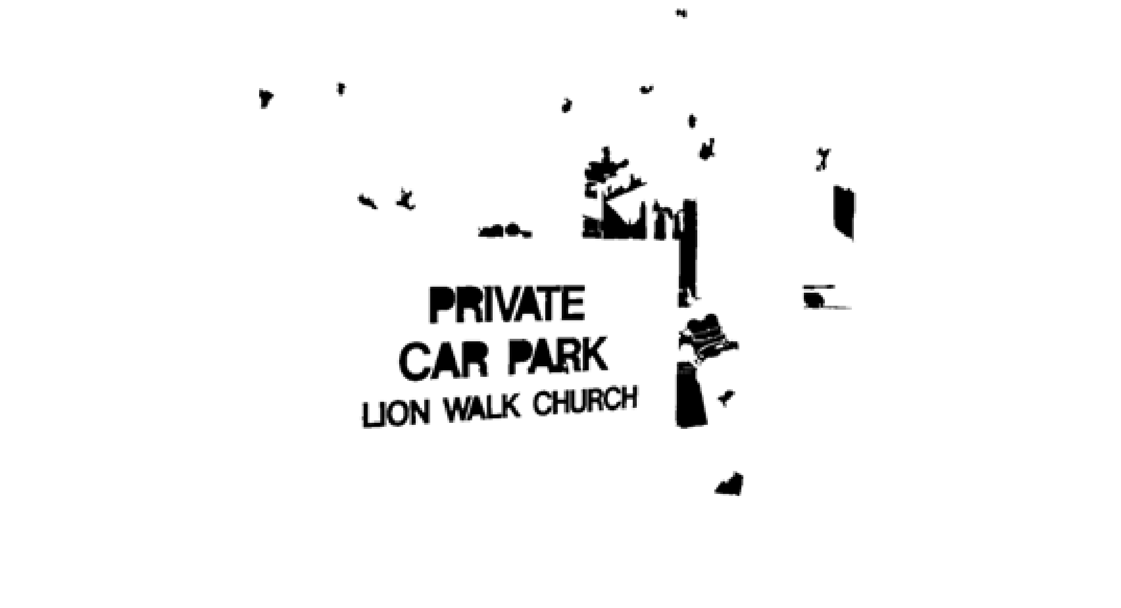


Figure 19: Final MSERS

4.3.3 Histogram of oriented gradient

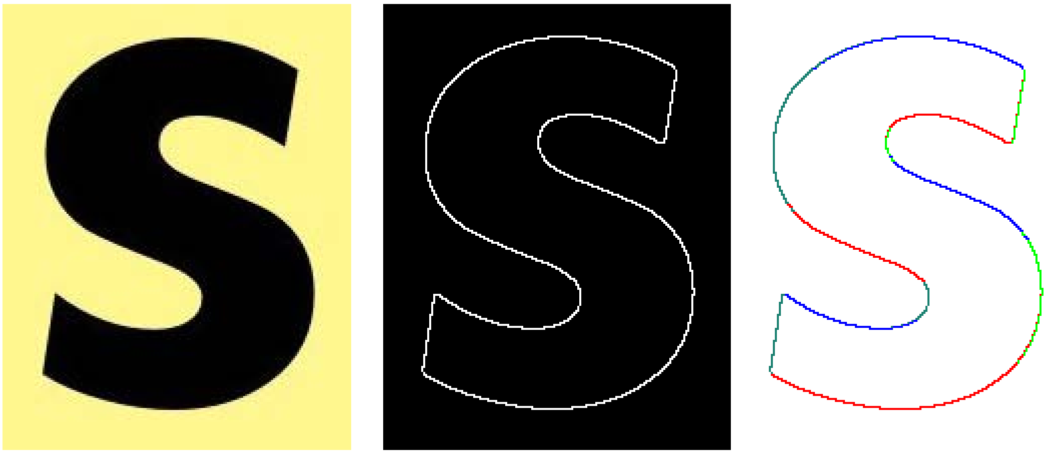
Now the bounding box of the MSERs selected in previous steps, are considered and the component from filtered edge image within a particular bounding box is extracted to make an edge sub-image of the corresponding MSER. Now we divide the ($0 \sim 2\pi$) into 4 bins and put edge pixels within a bin with respect to its gradient direction [12]. So points according to [12] in 4 bins are as follows

1) *PtType1:* $0 <\theta \leq\pi$ 4 or $7\pi/4 <\theta \leq 2\pi$;

2) *PtType2:* $\pi/4 <\theta \leq 3\pi/4$;

3) *PtType3:* $3\pi/4 <\theta \leq 5\pi/4$;

4) *PtType4:* $5\pi/4 <\theta \leq 7\pi/4$.



(a) **(b)** **(c)**

Figure 20: (a) Original image (b) Canny edge map (c) Points from 4 bin of oriented histogram. Reds are from PtType1, greens are from PtType2, blues are from PtType3, and sea-greens are from PtType4.

A text edge generally consists of all of these four kinds of edge points. More over edge in a text is a closed boundary and in pair. By pair what is meant is, for an edge there is almost another parallel edge. So as a matter of fact for an edge pixel there is another edge pixel at the parallel edge. These two pixels are called corresponding pixel and they has a relation with respect to their gradient direction. The distance between edge and its parallel counterpart or the distance between corresponding pixels is the stroke width. Any of this bin empty means the MSER is not a text MSER.

Corresponding pixel of PtType1 will be belonged to PtType3 and similarly corresponding pixel of PtType2 can be found in PtType4. In the Figure 20-(c) red pixel are from PtType1,green pixels are from PtType2, blue pixels are from PtType3 and the rest are from PtType4.So we can expect that number of pixel in PtType1 is nearly equal to number of pixel in PtType3. And number of pixel in PtType2 and number of pixel in PtType4 is nearly same for text candidates in the image. Now we also calculate ratio as

$$\boldsymbol{ratio1 = min\left(\frac{hist(PtType1)}{hist(PtType3)}, \frac{hist(PtType3)}{hist(PtType1)}\right)}$$

$$\boldsymbol{ratio2 = min\left(\frac{hist(PtType2)}{hist(PtType4)}, \frac{hist(PtType4)}{hist(PtType2)}\right)}$$

Where *hist* () is the number of points in that point type. For a text candidate $ratio1 > threshold$ and $ratio2 > threshold$. However rather using separate threshold on each of these ratios we multiply them and generate our first criterion feature.

$$\boldsymbol{decision1 = ratio1 * ratio2}$$

4.3.4 Stroke width consistency check

We will use this in unison with some other properties. Our next step is to calculate the stroke width variance and the consistency of stroke width. Stroke width is the distance between the pair edges or it is the distance between the corresponding pixels.

Consistency of stroke width means what portion of the edge pairs has a fixed stroke width. To calculate these two things rays are shot [6] from each edge pixel in a particular direction. We stop once we get another edge pixel of having nearly opposite gradient value (gradient value differ by almost *π*). Now length of this ray is one of the probable stroke widths. Now direction of that ray is something to be chosen judiciously, though magnitude is the gradient of that pixel. The direction of ray will be negative gradient for dark MSER on light background while it is positive for light MSER on dark background. Now why is it so? This can be demonstrated by the given Figure 21.

Both are image of same character. In one image '**S**' is in dark on light background and other one having 'S' in light on dark background. But the edge image for both the case will be nearly same. Note we use sobel operator to calculate the gradient of a pixel and the coordinate system in image is not the same thing we use our general practices. Note at the marked pixel in the white 'S' on black background *dy* is positive but *dx* is negative. So gradient is in *(π/2 ~ π)* interval anti-clockwise from positive *'x-axis'* and as per definition of gradient vector it is perpendicular with respect to edge direction (Figure 21-d). So moving towards this direction we eventually get an edge pixel (note Figure 21-e. the direction is depicted in green line). Now consider the second case of 'dark '**S**' on light background. Note here *dx* is positive but *dy* is negative which means the gradient direction is in *(3π/2~2π)* range (just opposite quadrant of previous case). Now if ray is shot in this direction a corresponding pixel will not be found. This direction is show in Figure 21-e by cyan arrow. So in this case rays are needed to shoot in negative direction. Thus a ray can be expressed as –

$$\boldsymbol{Q} = \boldsymbol{P} + \boldsymbol{n} * \boldsymbol{dir} * \boldsymbol{Grad}(\boldsymbol{P})$$

Where P is the starting point of the ray, *Grad(P)* is the gradient direction of P, dir is either 1 or -1 and n control the length of the ray. P and *Grad(P)* are both two-dimensional vector in this case.

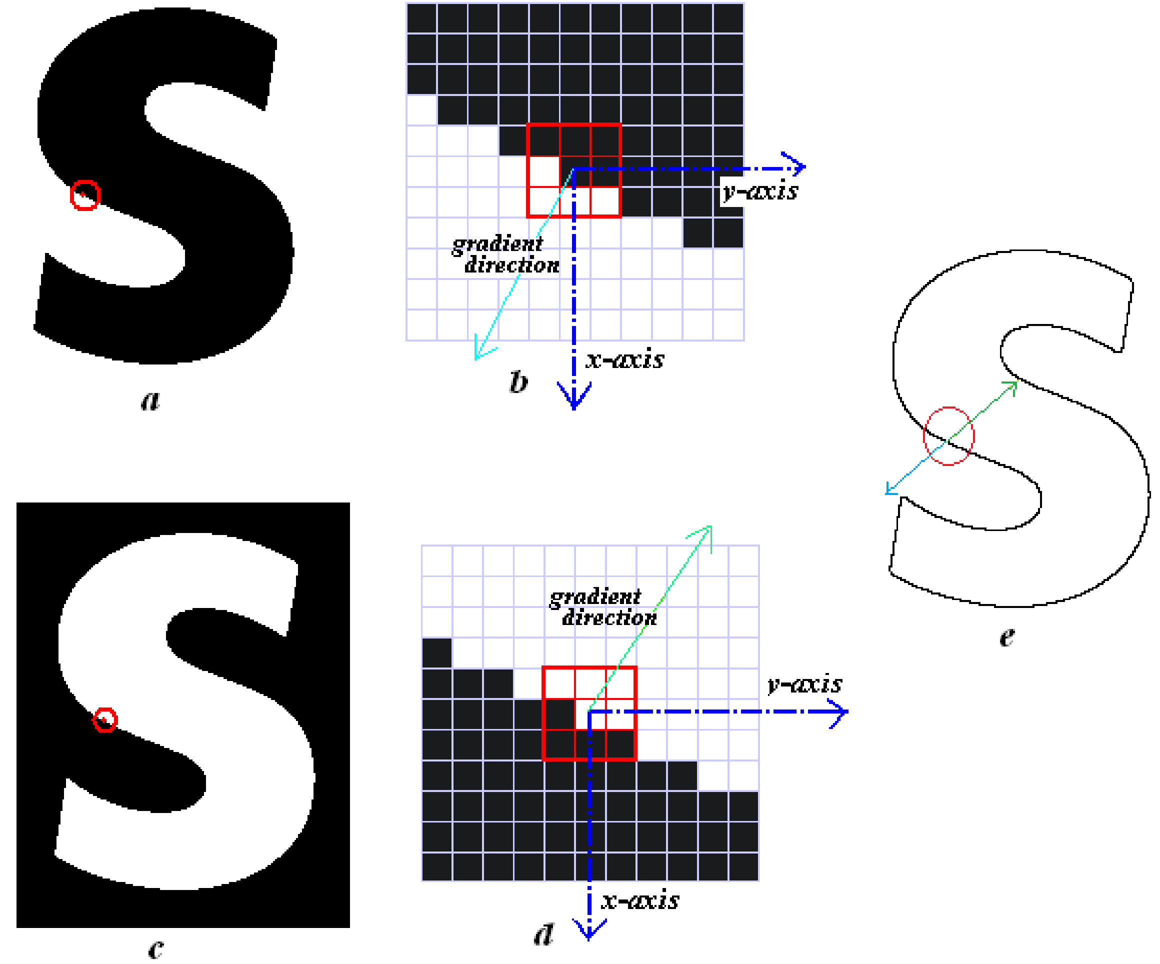


Figure 21: (a) Dark character on light background (b) gradient direction of marked pixel in a (c) bright character on dark back ground. (d) gradient direction of marked pixel in c. (e) gradient directions from both c and d.

Not each ray is considered in this ray shooting process. A ray proper in terms of its length with respect to height and width of the bounding box of the corresponding MSER is only considered. The angle of a ray is the angle of the straight line, which the ray consist of, with the horizontal axis. It is not the gradient angle of the pixel from which ray has been drawn, rather it is the inclination of that ray with respect to horizontal measured anti clockwise. If the ray makes an angle in the range *(0~ π/5)* is considered to be horizontal stroke and length of such should not exceed one third of the width of the bounding box. Similarly a stroke with angle in the range *(π/3.333 ~ π/2)* is considered to be vertical stroke and length should not go beyond one third of the height of the bounding box. A ray having angle within the range *(π/5 ~ π/2)* is checked with respect to both height and width. See the Figure17. Both the rays in green and cyan make same inclination with respect to horizontal and within *(0~ π/5)* though gradient angle of the cyan ray belongs to 4th quadrant.

Once we have all the eligible strokes we can calculate the statistical mean and standard deviation of the stroke widths and we can count the number of edge pixel for which corresponding stroke has a stroke width within 3-sigma limit. Ratio of this count to the total number of edge pixel of the concerned MSER is the consistency of stroke width. Now for a text candidate this consistency should be higher but the standard deviation should be lower.

We will describe shortly how these two approach are used in unison, however a third approach is needed where this first two fails to remove noise or about to delete valid text.

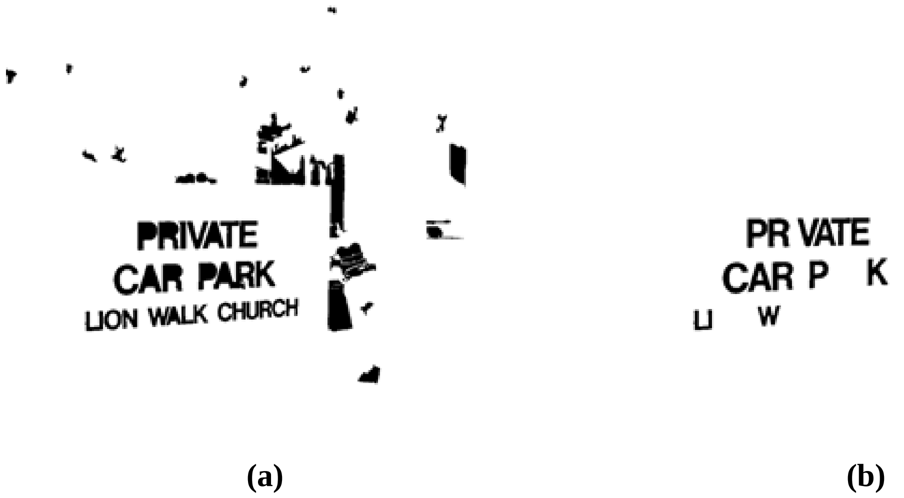
4.3.5 Strong edge Criteria

This approach is motivated from [5]. Let us have a brief look in to this interesting topic. In this method video frame is divided into 64 × 64 blocks. Now each block has undergone arithmetic mean filtering. Arithmetic mean filtering takes the average of intensity within a predefined window. Hence it introduces an amount of blurring in the whole image (block) and smoothes the noises as well as edge information. Median filtering is also done on the block. Median filtering keeps the median of intensity within a predefined widow. Hence less loss of intensity information. For each block AF (arithmetic mean filter) image and MF(median filtered) image is calculated. Difference image is obtained by subtracting the AF image from the MF image. For text blocks number of sobel edge element (NS_{AF}) in arithmetic mean filtered image is greater than the number of canny edge elements in the difference image (NC_{D_MA}). This is true because of the fact that the Sobel edge operator detects more edges when there is text information due to high sharpness in the text block. Therefore, the number of Sobel edge components in AF is greater than the number of Canny edge components in *D_MA*. Similarly, in case of non-text block the degree of blurring is very less in *D_MA* because the given block has no sharpness, in such cases the Canny detector detects some edges in *D_MA* but the Sobel detector does not detect edges in AF. Hence the number of Sobel edge components in AF is less than the number of Canny edge components in *D_MA* in case of non-text block.

A second rule in [5] deals with strong edge says that for text block number of strong edge will be more For a block difference of canny edge and sobel edge is the weak edge for that block. Now if we subtract the number of weak edge from the canny edge we get the number of strong edge. According to [5] for text blocks number of strong edge in MF ($\mathrm{NST}_{\mathrm{MF}}$) image is greater than number of strong edge in difference image (NST_{D_MA}). Figure 22 gives an idea how strong this strong edge criterion is. So direct use of this criterion may leads to loss of data. So this criterion will be used as second level help of some previously discussed criteria.

In this proposed method MSER bounding boxes are considered as blocks rather than dividing the actual image into blocks. So while operating on an MSER corresponding image segment from the gray scale image of input image is taken as block. The ratio r1 is calculated as $r_1 = NS_{AF} / NC_{D\_MA}$. The value of r_1>1 implies the rule1 in the paper [5]. The ratio r_2 calculated as $r_2 = NST_{MF} / NST_{D\_MA}$. So r_2 as per [5] should be more than 1.

However we find that is too strong to remove even some text candidates (see Figure 22). So it is better to keep lower limit of r_2 at 0.8.



(a) (b)

Figure 22: (a) Final selected MSER (b) MSER satisfying strong edge criterion

Now we are in a position to discuss how these things are put all together to remove noise but to preserve text elements.

4.3.6 Rule Set

Rule 1:

As per the properties of these parameters what is wanted, is-

1. Ratio1 and ratio2 of Section 4.3.3 should be nearer to 1 if not equal to 1.
2. Consistency of stroke width should be high.
3. Variance of stroke width should be less.

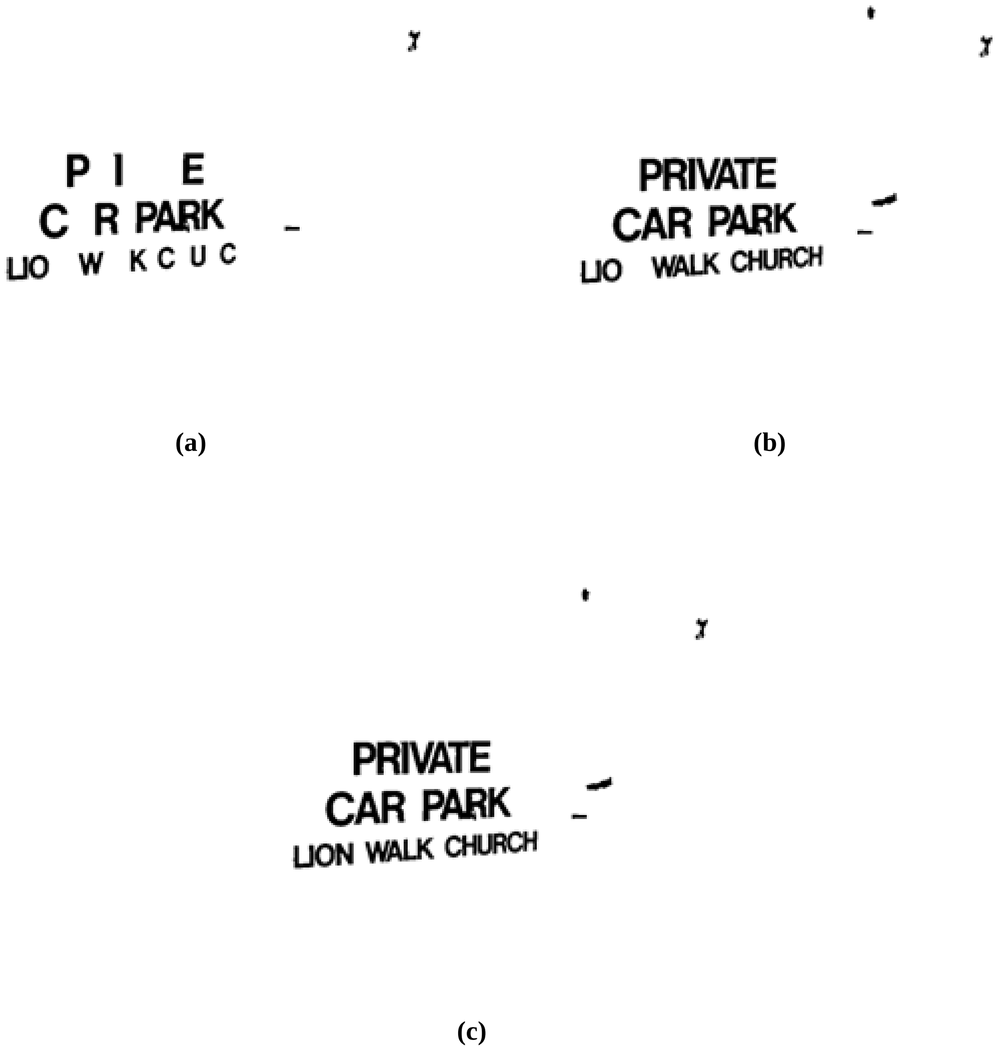
If these three criteria satisfied then we declare the MSER as text. Now separately ratio1 and ratio2 is assumed to have value greater than 0.7. Hence decision1 should be greater than 0.49. Now consistency of stroke width has to be greater than 0.55 and if variance is less than 5 then concerned MSER is a text MSER.

Rule 2:

There are some cases where this consistency of stroke width is not up to mark. This is because of size of MSER. If the MSER is small or due to filtering edge information is lost, in such case consistency of stroke width is low because of the fact that many edge pixel does not have their corresponding pixel. However there stroke width variance is really low. So to deal with we defined consistency should not be less than 0.3 and variance is less than 1.25. These values are obtained experimentally.

Rule 3:

In some cases consistency is within 0.3 to 0.55 but variance is greater than 1.25 and consistency is greater than 0.55 but variance is beyond 5. So handle such cases we need help of r1 and r2. In such cases we still have some bound for variance. In case of consistency within 0.3 to 0.55 we bound variance to be less than 3. Now for consistency beyond 0.55 variance must be less than a threshold defined as ratio of maximum of height and width of the bounding box of MSER to the probable stroke width. So we are allowing the variance to be as much as the dimension is greater than the stroke width. Now if the criteria satisfied then we check for the values r_2 (defined in *section 4.3.5*). Now if r_2 is lies within 0.8 to 4 then the MSER is a text MSER.



(a) (b)

(c)

Figure 23: (a) output after rule1 (b) output after rule1 and rule2 (c) output after rule1, rule2 and rule3

CHAPTER 5: Experimental results and Discussion

We implemented the proposed approach using MATLAB ver 13 under Windows environment. For evaluation purposes, we used a similar strategy as that of the ICDAR 2003 competition database [27]. As per ICDAR robust reading competitions a set of 3 measures is defined based on the ground tooth to evaluate a method. The definitions of these measures are given below-

5.1 Recall (R)

It is defines as the ratio of number of true detected blocks by the particular method to the number of all true text blocks presented in the ground tooth. It gives the idea how much of the actual blocks has been detected.

5.2 Precision (P)

It is the ratio of the number of truly detected blocks by the particular method to the number of all detected blocks by the method. It gives the idea about what portion of all detected blocks by the method are actual text blocks.

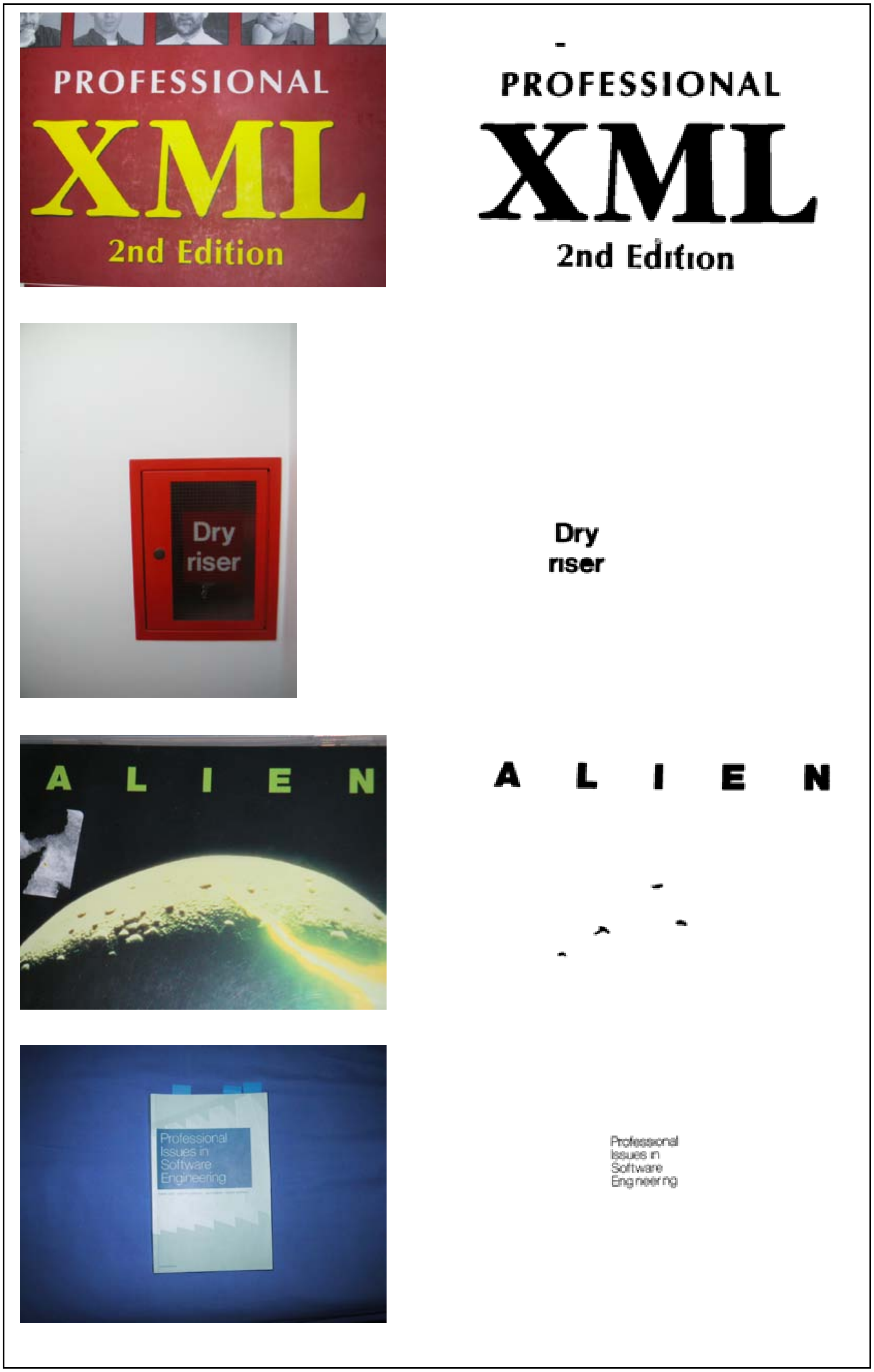
5.3 F-measure (F)

It is defined as-

$F\text{-}measure = 2 \times P \times R / (P+R)$.

Algorithm	Recall	Precision	F-measure
Epshteon.et.al[6]	0.73	0.60	0.66
Shivakumara.et.al[11]	0.86	0.82	0.84
Huang .et. al[7]	.6377	.6198	.6286
Proposed Method	***0.783***	***0.84***	***0.785***

Table 1: Comparative study of the proposed method


PROFESSIONAL
XML
2nd Edition
PROFESSIONAL
XML
2nd Edıtıon
Dry
riser
Dry
rıser
A L I E N
A L I E N
Professional
Issues in
Software
Engineering
Professıonal
Issues ın
Software
Engneerng

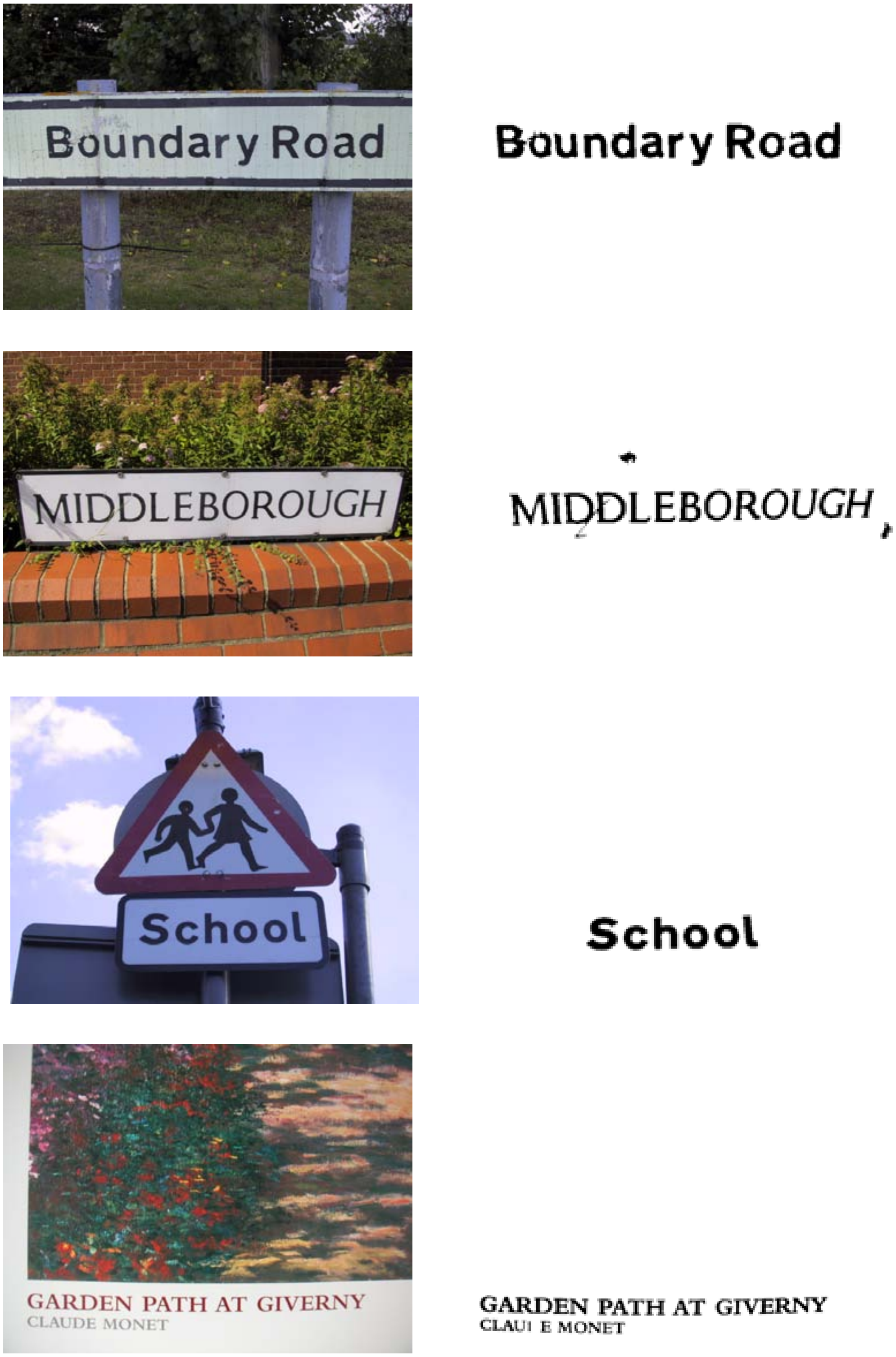


Figure 24: actual images and their output

5.4 Output on scene image containing texts of multiple scripts

Since the proposed method did not use any script specific feature, it should be able to extract texts of any script or language from input image. This claim is validated by simulating the proposed approach on local scene images containing texts of at least two of Devanagari, Bangla and English. Figure 25 shows outputs on a few such images.

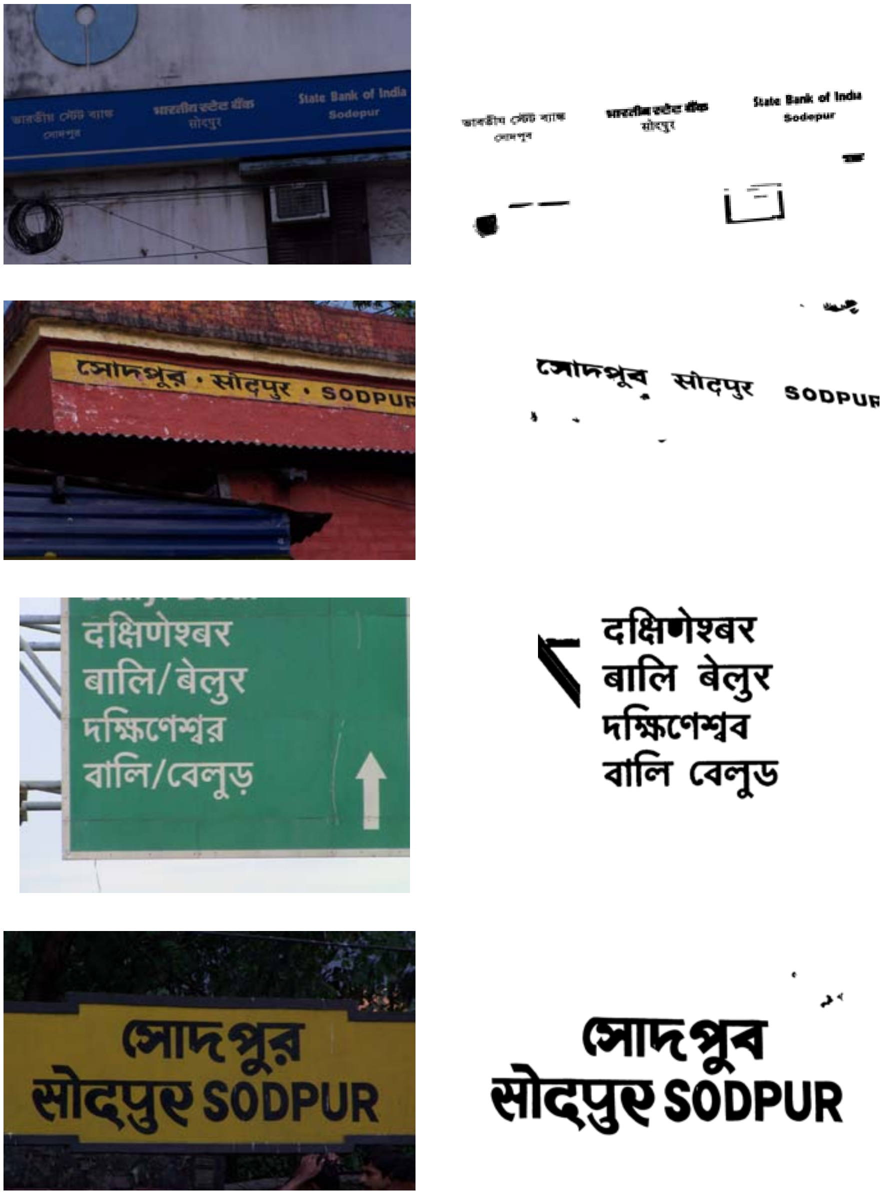
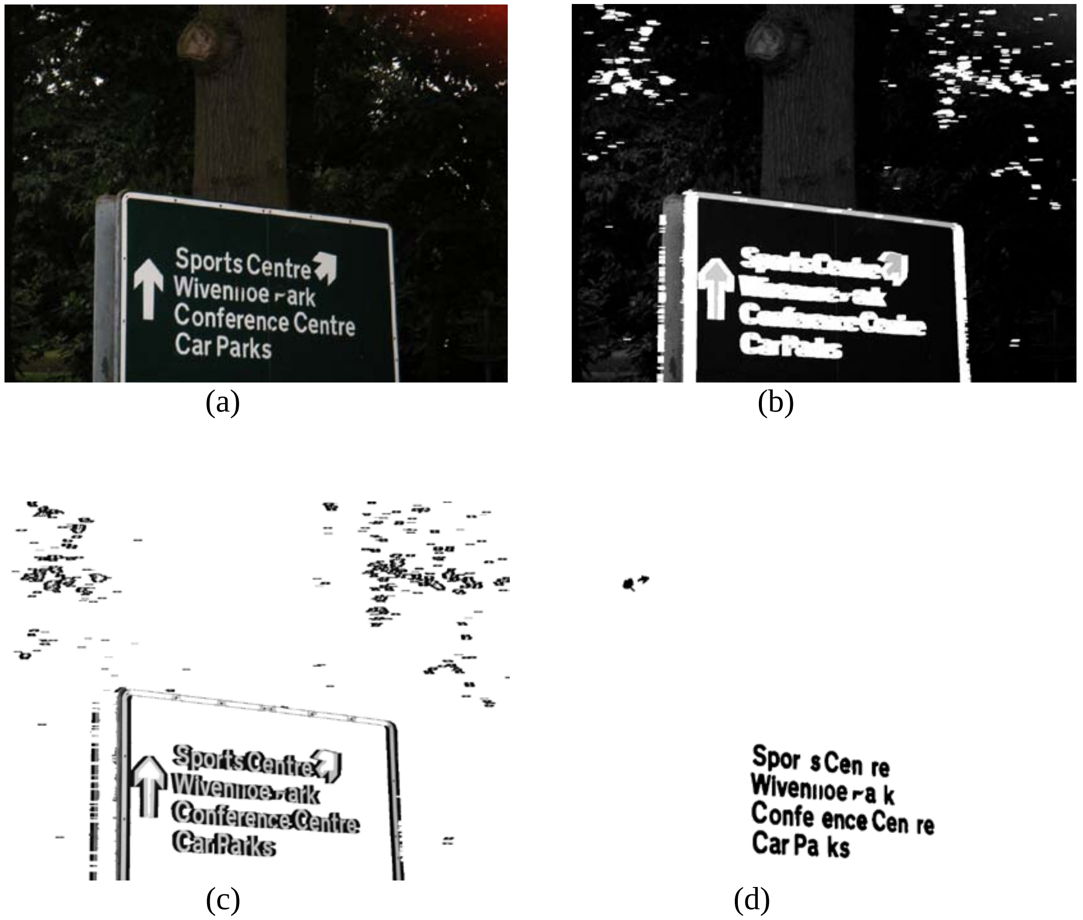


Figure 25: output on Devnagari / Bangla script

5.5 Discussions

The proposed method is compared with some existing standard method in table1. It is found to be superior by *precision* over all other methods. However it is just below in case of *recall* and *f-measure* by the method proposed in [11]. As discussed in literature study this method [11] at some stage uses maximum difference, “defined as the difference between the maximum value and the minimum value within a local 1 × N” window. According to Wong and Chen [26], this value of N should be a little larger than the stroke width of the largest character wanted to be detected. Now this stroke width information cannot be known in prior and with a fixed value of N the method can miss out all other



(a) (b)

(c) (d)

Figure 26: (a) original image (b) partial output by [11] with N=19 (c) output by [11] with N=21 (d) output by out method

text which has stroke width greater than this fixed value of N. We have implemented this method up to K-means clustering to have possible text blocks. Here a sample of output is presented in Figure 26.

5.6 Limitations

The challenges of this work is not only in the task of eliminating noises but some time text in the images appeared with some ornamental shape with graphic designs (Figure 27), which is really hard to detect without some contextual knowledge.

Another problem is unequal luminance over the scene. Sometime this misbalance is created by camera flash light. Some part of the image is so much lighten that MSER is not properly detected. As a matter of fact text information gets lost.

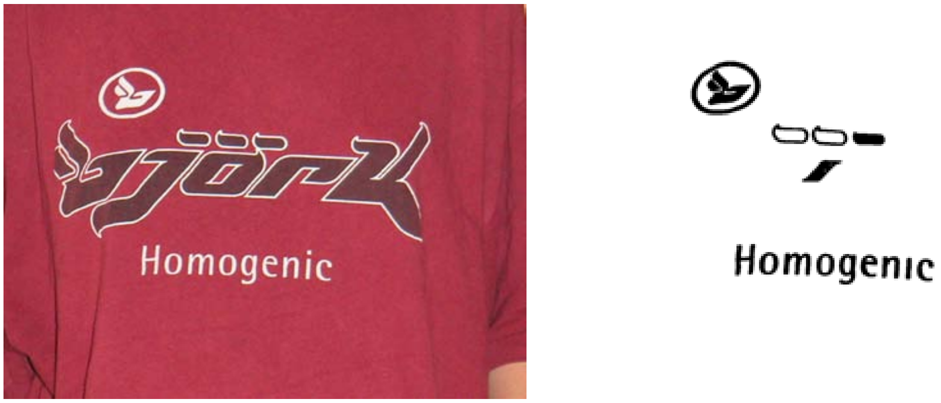


Figure 27: graphic text are not detected

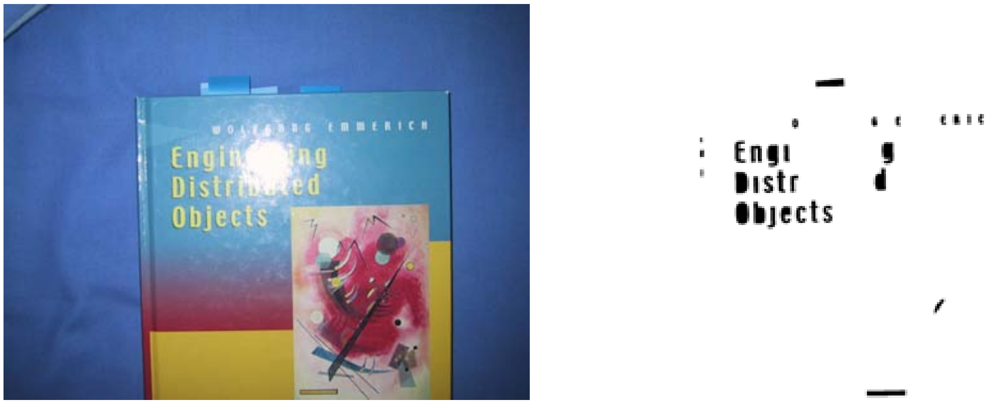


Figure 28: uneven illuminations creating problem

If the text in the scene is too small then there is a probability for their corresponding edges to be deleted in the edge image by filtering. Or they are small enough not to produce edges in perfect pairs.

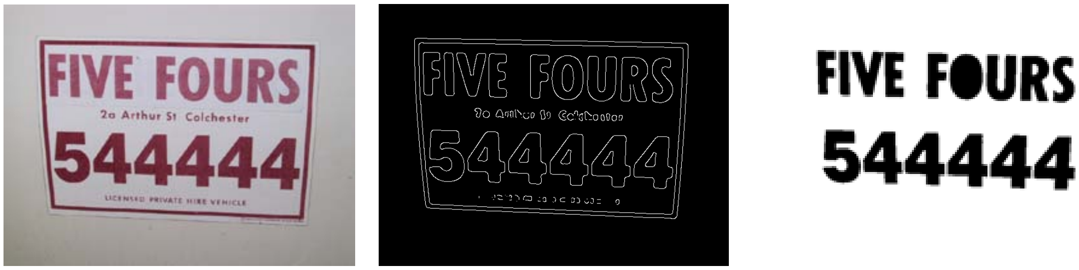


Figure 29: Deficiency of edge pixel

But there are some situations where noise really disturbs a lot. These noises are of regular shapes and satisfy all the necessary property discussed in this thesis to be declared as a text.

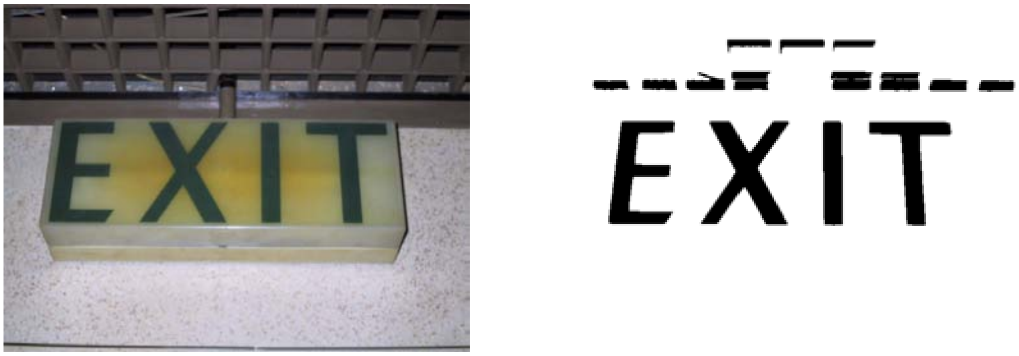


Figure 30: Noises getting detected

5.7 Test on video data

Though at it first place it is not designed for extraction of texts from video frames, but here are same example of output on video frame data.

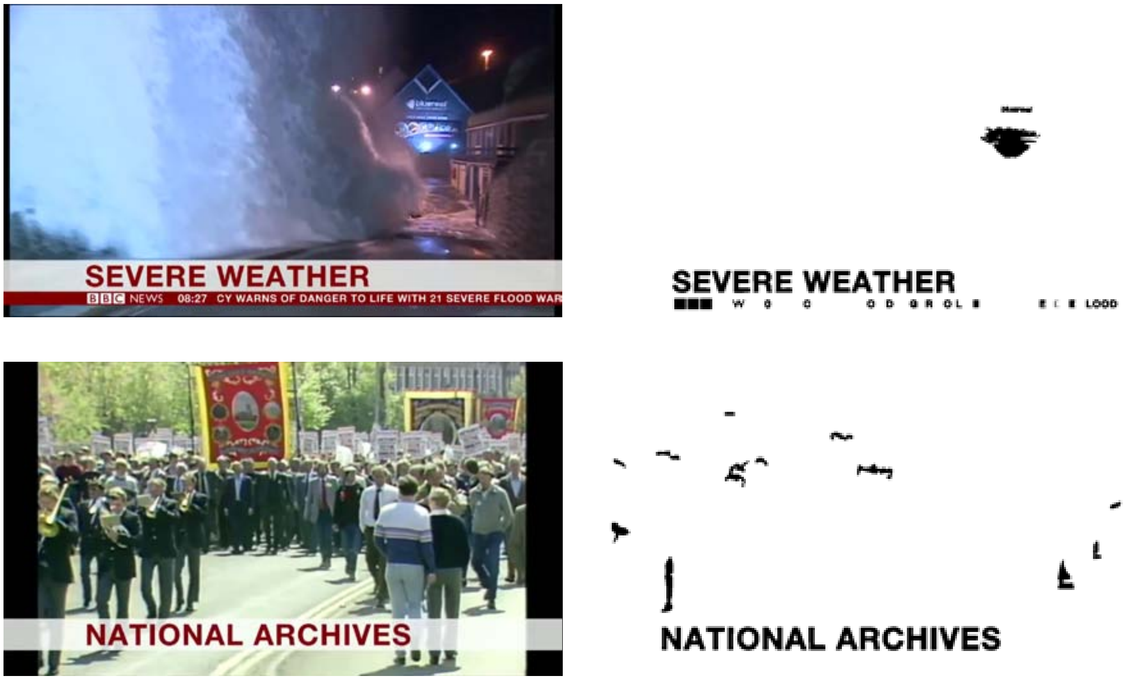

SEVERE WEATHER
BBC NEWS 08:27 CY WARNS OF DANGER TO LIFE WITH 21 SEVERE FLOOD WAR
SEVERE WEATHER
NATIONAL ARCHIVES
NATIONAL ARCHIVES

Figure 31: output on video frames

5.8 Future Scope

The limitations of the method open the door for its future scope. As discussed unequal laminations, non-prominent text, and noises etc makes the work challenging. The approaches which can be incorporated to improve the performance can be summarized as follows-

- Some contextual information can be used to retrieve the text MSERS which are compromised as noises. As for example text in scene images are generally occurs in group. It may be so that a character of a text group detected as noise. Now from its neighbor information and similarity in height, width color, and orientation this character can be saved from getting deleted. In [4] an approach called link energy has been discussed.

- Another attack to this problem can be done associating dictionary into the scenario. For that to happen, the part of the text has to be recognized. Then from a dictionary of corresponding language by shortest editing distance the near most word can be found.

- As this method uses Gaussian pyramid at it initial phase and at each level the computation is not depend on the other stages except at the last when all are combined. So there is a scope to introduce parallelism in the method for speed up the whole procedure.

5.9 Conclusion

The motive of thesis was to build a system to extract text from the scene images. The procedure here discussed is unsupervised and based on some general properties that are not language specific. Still it is giving better recall than some existing methods and superior by precisions. More over this method is not script specific. Though sometime some character part gets missed due to some rule get harder over them; some time noises rules over texts; some time limitation of MSER detector is being a bar but nevertheless there are always scope of prospect and future work as discussed above. Context related information; neighborhood study can retrieve some of missing text. But in its scope and simplicity this method is good enough at accomplishing its goal. Moreover it is able to segregate the background information of text and extract the text. So this output can be fed to OCR. When OCR comes into picture then dictionary based searching by ‘Edit distance‘ or ‘Levenshtein distance’ can help to retrieve more information and attain more accuracy. This extracted text can be recognized by OCR then can be converted to corresponding speech for further advancement of scope of this work.

Appendix A: Gaussian Pyramid

Gaussian pyramid [2] is a well known method of decomposition an image in multiple scales. They are widely used in image coding, image blending, image enhancement and just too many other applications.

Let the initial image be g_0 which contains C columns and R rows of pixels. This one is the lowest level image in the pyramid. Now g_1, level one image of the pyramid, is generated from g_0 by low pass filtering and sub-sampling the image g_0. Each value in level-1 is the weighted average of values in level-0 within a 5×5 window. Similarly each upper layer of pyramid is generated using the pixels values of the lower layer and same pattern of weights.

$$g_l(i,j) = \sum_{m=-2}^{2} \sum_{n=-2}^{2} w(m,n) g_{l-1}(2i+m, 2j+n)$$

Where $0 < l < N$ and $0 \leq i \leq C_l$ and $0 \leq j \leq R_l$ Where N is the number of levels in the pyramid. C_l and R_l is the number of column and rows in the image at l-th level of the pyramid.

The weighting kernel is chosen to be separable for simplicity.

$$w(m,n) = w(m)w(n)$$

Moreover one dimensional, length 5, function w is normalized and symmetric and have a constraint called equal contribution. So

$$w(0) = a; w(1) = w(-1) = \frac{1}{4}; w(2) = w(-2) = \frac{1}{4} - \frac{a}{2}$$

We chose $a = 0.375$.

This function of sub-sampling and low-pass filtering over the level of pyramid is often called *reduce*. So for a level *l* and *l-1*.

$$g_l = reduce(g_{l-1})$$

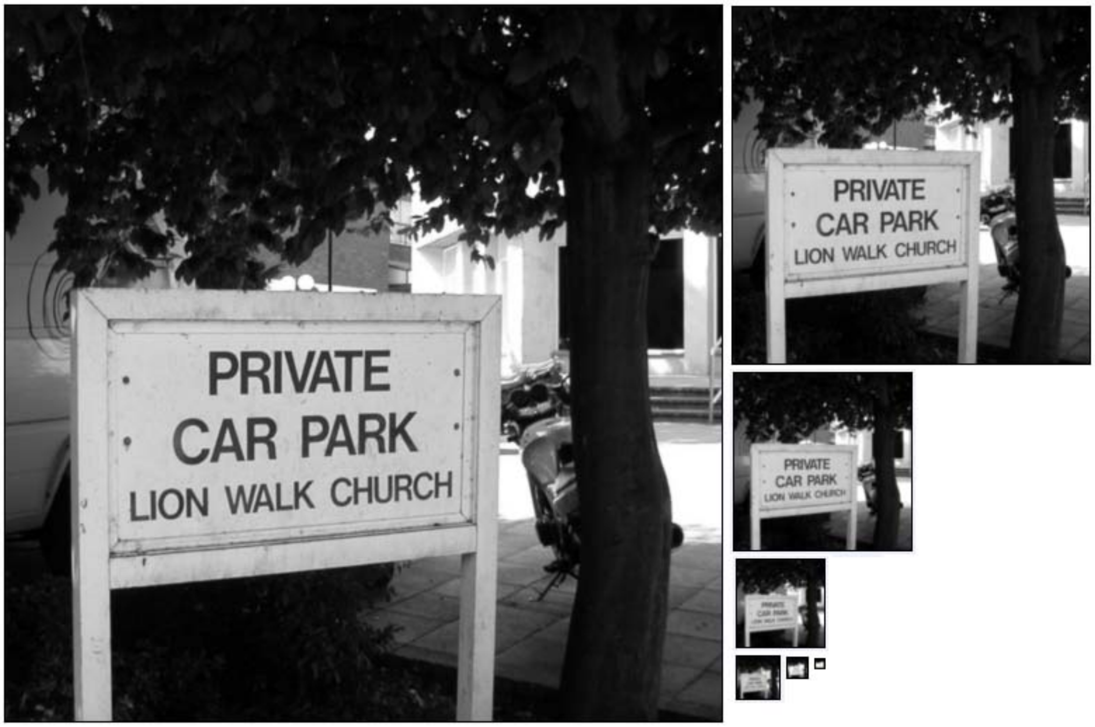


Figure 32: images over the level of Gaussian pyramid

there is another function expand for Gaussian pyramid Interpolation. Use of this function is very prominent in case of Laplacian pyramid. However detail discussion about Laplacian pyramid [2] is out of the scope of this thesis.

Appendix B: MSER

MSER stands for maximally stable extremal region. At its crux extremal region are those connected component in images which remain unchanged under monotonic change in intensities of images. Secondly it preserved topology of pixels under continuous geometric transformations [14]. In other words extremal regions are those regions of which all pixels have either greater or lower intensity than those which surrounds the region but does not belong to the ER. Maximal stable extremal regions are those ER which preserves their stability over a range of thresholding. The set of extremal regions *E*, i.e., the set of all connected components obtained by thresholding, has a number of desirable properties. Firstly, a monotonic change of image intensities leaves *E* unchanged, since it depends only on the ordering of pixel intensities which is preserved under monotonic transformation. This ensures that common photometric changes modelled locally as linear or affine leave *E* unaffected, even if the camera is non-linear (gamma-corrected). Secondly, continuous geometric transformations preserve topology– pixels from a single connected component are transformed to a single connected component. Thus after a geometric change locally approximated by an affine transform, homography or even continuous non-linear warping, a matching extremal region will be in the transformed set *E*. Finally, there are no more extremal regions than there are pixels in the image. So MSER can be of two types dark MSER on light back ground and bright MSER on dark background.

Mathematical definition of MSER can be found in [1]. Here a snapshot of mathematical definitions from [1] is furnished below

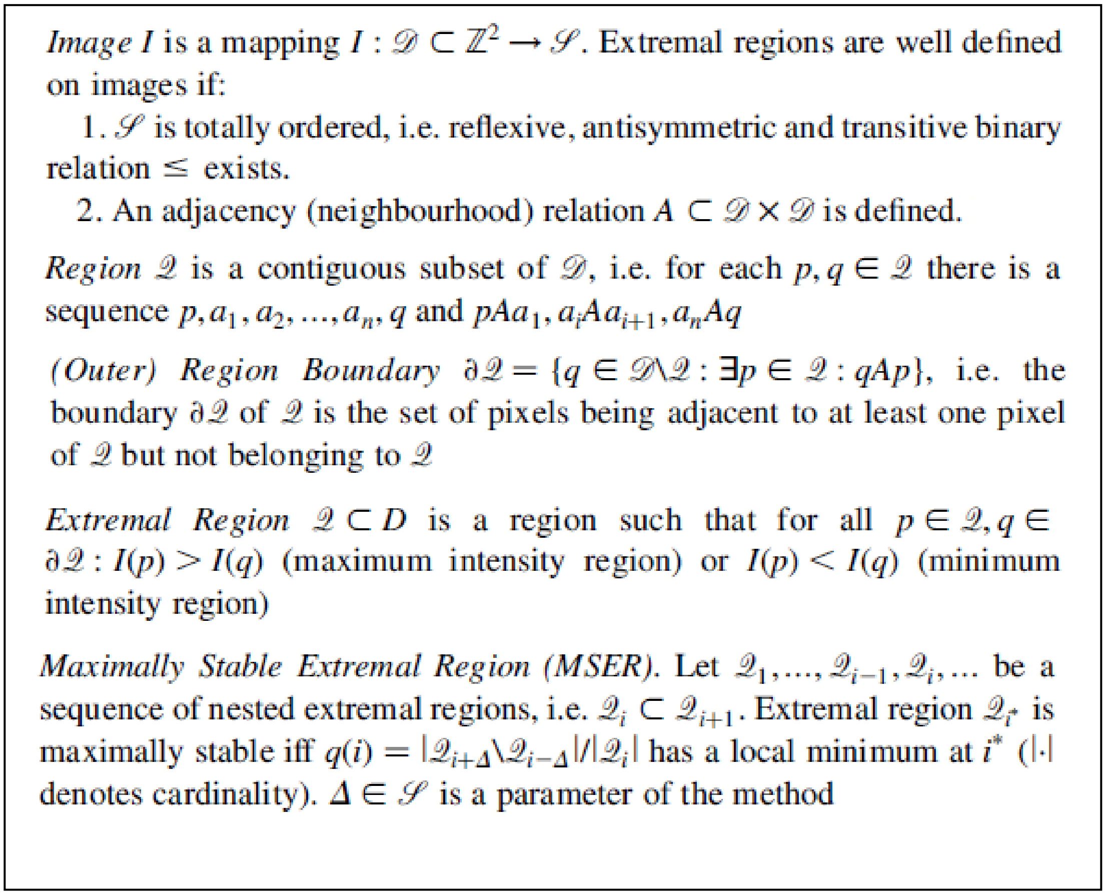


Image I is a mapping $I : \mathcal{D} \subset \mathbb{Z}^2 \rightarrow \mathcal{S}$. Extremal regions are well defined on images if:

1. $\mathcal{S}$ is totally ordered, i.e. reflexive, antisymmetric and transitive binary relation $\leq$ exists.
2. An adjacency (neighbourhood) relation $A \subset \mathcal{D} \times \mathcal{D}$ is defined.

Region $\mathcal{Q}$ is a contiguous subset of $\mathcal{D}$, i.e. for each $p, q \in \mathcal{Q}$ there is a sequence $p, a_1, a_2, \ldots, a_n, q$ and $pAa_1, a_iAa_{i+1}, a_nAq$

(Outer) Region Boundary $\partial\mathcal{Q} = \{q \in \mathcal{D} \backslash \mathcal{Q} : \exists p \in \mathcal{Q} : qAp\}$, i.e. the boundary $\partial\mathcal{Q}$ of $\mathcal{Q}$ is the set of pixels being adjacent to at least one pixel of $\mathcal{Q}$ but not belonging to $\mathcal{Q}$

Extremal Region $\mathcal{Q} \subset D$ is a region such that for all $p \in \mathcal{Q}, q \in \partial\mathcal{Q} : I(p) > I(q)$ (maximum intensity region) or $I(p) < I(q)$ (minimum intensity region)

Maximally Stable Extremal Region (MSER). Let $\mathcal{Q}_1, \ldots, \mathcal{Q}_{i-1}, \mathcal{Q}_i, \ldots$ be a sequence of nested extremal regions, i.e. $\mathcal{Q}_i \subset \mathcal{Q}_{i+1}$. Extremal region $\mathcal{Q}_{i^*}$ is maximally stable iff $q(i) = |\mathcal{Q}_{i+\Delta} \backslash \mathcal{Q}_{i-\Delta}| / |\mathcal{Q}_i|$ has a local minimum at i^* ($|\cdot|$ denotes cardinality). $\Delta \in \mathcal{S}$ is a parameter of the method

As discussed in very beginning of chapter 4 if tree is build by the connected component at each thresholding level and children of a connected component are those connected components in next level which get separated due to increase in value of threshold. This tree structure is well described in [17]. Here also a sample tree structure is presented in Figure 33. The left side corresponds to the tree of connected component and right side image is the whole image at that thresholding level, and each level is annotated with the corresponding gray level.

The stability value for a region R_i is defined as per [17] -

$$\Psi = \frac{(|R_j^{g-\Delta}| - |R_k^{g+\Delta}|)}{|R_i^g|}$$

Where R_i^g is a region at gray level g and Δ is a stability range parameter. |.| is set cardinality operator. $R_j^{g-\Delta}$ and $R_k^{g+\Delta}$ are extremal regions found moving upward and downward respectively from R_i^g along the tree path until a region with gray value g-Δ and g+Δ is found. MSERs correspond to those nodes of the tree that have a stability value Ψ which is local minimum along the path to the root of the tree.

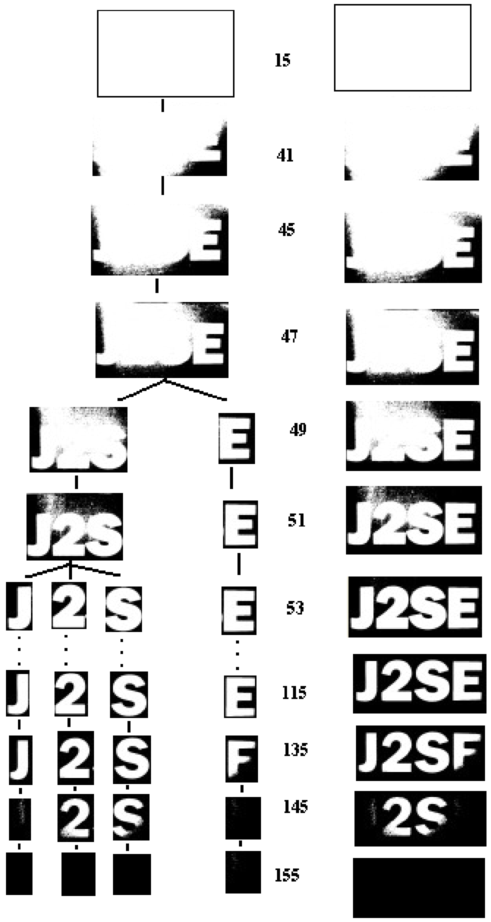


Figure 33: sample ER component tree

References

[1] J. Matas, O. Chum, T. Pajdla, Robust wide-baseline stereo from maximally stable extremal regions, Image And Vision Computing, vol. 22 (10), pp. 761-767, 2004.

[2] P. J. Burt, E. H. Adelson, The Laplacian Pyramid as a Compact Image Code, IEEE Transactions on Communication, vol. 31(4), pp. 532–540, 1983.

[3] W. Mao, F. Chung, Kenneth. K. M. Lam, W. Siu, Hybrid Chinese/English Text Detection in Images and Video Frames, Procs. of the 16th Int. Conf. on Pattern Recognition (ICPR'02), vol. 3, pp. 310-315, 2002.

[4] J. Zhang, R. Kasturi, Character Energy and Link Energy-Based Text Extraction in Scene Images, Procs. of the 10th Asian conference on Computer vision (ACCV'10), vol. 6493, pp. 308-320, 2011.

[5] P. Shivakumara, H. Weihua, C. L. Tan, An Efficient Edge based Technique for Text Detection in Video Frames Document Analysis Systems, Proc. the 8th Int. Workshop on Document Analysis Systems, 2008.

[6] B. Epshtein, E. Ofek, and Y. Wexler, Detecting Text In Natural Scenes With Stroke Width Transform, Proc. of CVPR, pp. 2963-2970, 2010.

[7] H. Rong, P. Shivakumara, S. Uchida, Scene Character Detection by an Edge-Ray Filter, Proc. of 12th Int. Conf. on Document Analysis and Recognition (ICDAR), pp. 462-466, 2013.

[8] C. Yao, X. Bai,; W. Liu, Yi Ma, Z. Tu, Detecting texts of arbitrary orientations in natural images, Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , pp. 1083-1090, 2012.

[9] C. Shi, C. Wang, B. Xiao, Y. Zhang, S. Gao, Scene text detection using graph model built upon maximally stable extremal regions, Pattern Recognition Letters, Volume 34, Issue 2, 15 January 2013, Pages 107-116, ISSN 0167-8655

[10] Liu.C, Wang.C, Dai.R; Text Detection in Images Based on Unsupervised Classification of Edge-based Features ICDAR '05 Proc. of the 8th Int. Conf.on Document Analysis and Recognition ,pp 610-614.

[11] P. Sivakumara, T. Q. Phan, C. L, Tan, A Laplacian Approach to Multi-Oriented Text Detection in Video, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33(2). Pp. 412-419, 2011.

[12] J. Zhang and R. Kasturi. Text Detection using EdgeGradient and Graph Spectrum, Proc. of the 20th Int. Conf.on Pattern Recognition, pp.3979-3982, 2010.

[13] Aruni Roy Chowdhury, Ujjwal Bhattacharya, Swapan K. Parui: Text Detection of Two Major Indian Scripts in Natural Scene Images. CBDAR 2011: PP 42-57.

[14] K.Mikolajczyk, T. Tuytelaars, C.Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T.Kadir, L. Gool , A comparison of affine region detectors. Int. J. Computer Vision 65 (1), 43–72, 2005.

[15] G. Borgefors, “Distance transformations in digital images,” Computer Vision, Graphics and Image Processing, vol. 34, pp. 344-371, 1986.

[16] G.R.Bradski, Real time face and object tracking as a component of a perceptual use interface. In proc. IEEE Workshop on Applications of Computer Vision 1998.

[17] M. Donoser, H. Bischof, "Efficient Maximally Stable Extremal Region (MSER) Tracking," Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on , vol.1, no., pp.553,560, 17-22 June 2006

[18] Keechul Jung, Kwang In Kim, Anil K. Jain, Text information extraction in images and video: a survey, Pattern Recognition, Volume 37, Issue 5, May 2004, Pages 977-997, ISSN 0031-3203

[19] R.Gao, F. Shafiat, S. Uchida, Y. Feng, Saliency inside Saliency – A Hierarchical Usage of Visual Saliency for Scene Character Detection, CBDAR 2013.

[20] H. Wang, Y. Landa, M. Fallon, S. Teller, Spatially Prioritized and Persistent Text Detection and Decoding, CBDAR 2013.

[21] J. Gllavata, R. Ewerth, B. Freisleben, "Text detection in images based on unsupervised classification of high-frequency wavelet coefficients,". Proc. of the 17th Int. Conf. on Pattern Recog. , vol 1, pp.425-428, 2004

[22] J. Ohya, A. Shio, S. Akamatsu, Recognizing characters in scene images, IEEE Trans. Pattern Anal. Mach. Intell. 16 (2) (1994) 214–224.

[23] Y. Zhong, K. Karu, A.K. Jain, Locating text in complex color images, Pattern Recognition 28 (10) (1995) 1523–1535.

[24] C.M. Lee, A. Kankanhalli, Automatic extraction of characters in complex images, Int. J. Pattern Recognition Artif. Intell. 9 (1) (1995) 67–82.

[25] Q. Ye, D. S. Doermann, Scene Text Detection via Integrated Discrimination of Component Appearance and Consensus, Proc. Of CBDAR, pp. 47-59, 2013.

[26] E. K. Wong and M. Chen, “A New Robust Algorithm for Video Text Extraction,” Pattern Recognition, vol. 36, pp. 1397-1406, 2003.

[27] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong and R. Young, “ICDAR 2003 Robust Reading Competitions,” Proc. of the 7th Int. Conf. on Doc. Anal. and Recog., pp. 682-687, 2003.