Text Detection in Videos Using Super Resolution

A DISSERTATION
submitted towards the fulfillment of the
requirement for the award of the degree of
MASTER OF TECHNOLOGY
in
Computer Science and Engineering

By

MOHD DANISH

Department of Computer Science and Engineering
INDIAN INSTITUTE OF TECHNOLOGY, ROORKEE
Roorkee - 247667, India

JUNE 2016

AUTHOR'S DECLARATION

I declare that the work presented in this dissertation with title **"Text Detection in Videos Using Super Resolution"** towards the fulfillment of the requirement for the award of the degree of **Master of Technology** in **Computer Science & Engineering** submitted in the **Department of Computer Science & Engineering, Indian Institute of Technology Roorkee, India** is an authentic record of my own work carried out during the period from **June 2015 to May 2016** under the supervision of **Dr. Partha Pratim Roy**, Assistant Professor, Department of Computer Science and Engineering, Indian Institutes of Technology, Roorkee and **Dr. Debashis Sen**, Assistant Professor, Department of Electronics & Electrical Communication Engineering, Indian Institutes of Technology, Kharagpur. The content of this dissertation has not been submitted by me for the award of any other degree of this or any other institute.

DATE: .... SIGNED: ....
PLACE: .... (MOHD DANISH)

CERTIFICATE

This is to certify that the statement made by the candidate is correct to the best of my knowledge and belief.

DATE: .... SIGNED: ....

(DR. PARTHA PRATIM ROY)
Assistant Professor
Indian Institutes of Technology, Roorkee

Abstract

Text in images and videos contain useful information for automatic annotation, indexing, and structuring of images and videos. Extraction of this information involves detection, localization, tracking, extraction, enhancement, and recognition of the text from a given image and videos. However, variations of text due to differences in size, style, orientation, and alignment, as well as low image contrast and complex background make the problem of automatic text extraction extremely challenging.In the complete process of text extraction, text detection is the primary and fundamental step. We can increase the performance of the text detection by giving high-resolution images or video frames instead of low-resolution images or video frames. Here, Super-resolution can be used to produce the high-resolution image or video frame. In high-resolution images or video frames, there is more pixel density and thus provides finer details of the image or scene. In this report, we talk about how we can increase the range of different size text detected from videos using super resolution. More detected text from a video will imply performance increase in text recognition.

Dedication and acknowledgements

Firstly , I would like to express my deepest gratitude to my supervisor, Dr. Partha Pratim Roy, for his excellent guidance, meaningful insights and moral support. He has been supportive since the day I began working on this dissertation and gave me the freedom I needed to explore this area of research on my own, while pointing me in the right direction in the times of need. His comprehensive knowledge in the area of computer vision & machine learning and hard working nature has been a constant source of inspiration.

My co-supervisor, Dr. Debashis Sen, has been always there to listen and give advice. His insightful comments and constructive criticisms at different stages of my research were thought-provoking and they helped me focus my ideas. I am grateful to him for holding me to a high research standard and enforcing strict validations for each research result, and thus teaching me how to do research.

I am also grateful to the Dept. of Computer Science, IIT-Roorkee for providing valuable resources to aid my research.

Finally, hearty thanks to my parents and siblings, who encouraged me in good times, and motivated me in the bad times, without which this dissertation would not have been possible.

TABLE OF CONTENTS

Page

List of Tables vii

List of Figures viii

1 Introduction 1

1.1 Fundamental Sub Problems in Text Information Extraction 4

1.1.1 Text Detection 4

1.1.2 Tracking, extraction, and enhancement 5

1.1.3 Text recognition 5

1.2 Enhancement of Text Using Super Resolution 5

1.2.1 What is Super Resolution? 5

1.2.2 Image Registration 6

1.2.3 Algorithms for Super Resolution 8

1.3 Dataset Description 10

1.4 Overview of the problem 12

2 Related Work 13

2.1 Some Basic method of text detection in images and videos 13

2.2 Text detection in images and videos using neural networks 15

2.3 Text detection in images and videos using MSERs 15

3 Proposed Work 18

3.1 Extraction of Key Frames from Videos 18

3.2 Applying Super Resolution 20

3.3 Detect MSERs regions 21

3.4 Mapping HR image and LR Image 22

3.5 Using a classifier to separate text and non text regions 23

3.5.1 Feature Extraction 23

3.5.2 Classification 25

4 Experiment Results 26

4.1 Text Detection Results . . . 26
4.2 Problems Associated with Text Detection . . . 28
4.3 Classification Results of Text and Non-Text using Feature Extraction . . . 29
4.3.1 Size of HOG features . . . 29
4.3.2 Size of LBP features . . . 30
4.3.3 Analysis of Results of Classifiers . . . 31
4.4 Classification Results of Text and Non-Text using Convolutional Neural Network . . . 36

5 Conclusion and Future Work **41**

References **42**

List of Tables

Table **Page**

1.1 Detailed description of videos from ICDAR 2015 11
1.2 Detailed description of videos from IITR videos dataset 11

2.1 Accuracy on different dataset of method proposed by A K Jain et al. 14

4.1 Confusion Matrix for Naive Bayes Classifier 33
4.2 Confusion Matrix for SVM Classifier 34
4.3 Confusion Matrix for KNN Classifier 34
4.4 Precision,recall and F-score values for classification of Text and Non-Text regions 35
4.5 ConvNet Variations 39
4.6 Precision, Recall, F-score and Accuracy of above CNN models 40

List of Figures

Figure **Page**

1.1 Figure showing variations of text in images from MSRA-TD500 dataset . . . 2
1.2 Figure showing variations of text in video frames from ICDAR dataset 3
1.3 Architecture of the TIE System 4
1.4 2-D representation of pixels in images taken at sub pixel level 6
1.5 btained high resolution image by registering four lower resolution images at a sub-pixel level 7
1.6 Details of non-uniform interpolation 8
1.7 Retrieved patches from database for an input low resolution patch 10

2.1 Overview of method proposed by A K Jain et al. 14
2.2 Overview of text detection scheme prorposed by H Li et al. 15
2.3 architecture of whole process proposed by H Li et al. 16
2.4 Flowchart of algorithm proposed by C Shi et al. 16
2.5 Illustration of process proposed by H Chen et al. 17

3.1 Outline of the proposed method 19
3.2 Figure showing robustness of MSERs text detector against view point 22
3.3 Figure showing robustness of MSERs text detector against view point 23
3.4 Figure showing robustness of MSERs text detector against view point 24
3.5 Mapping Detected Regions from HR Image onto LR Image. 24

4.1 Text Detection Example 26
4.2 Comparison of text detection in LR image and HR image of key frame shown in figure 4.1a 27
4.3 Comparison of text detection in LR image and HR image of another key frame 27
4.4 Text Detection Results on ICDAR dataset 28
4.5 Text Detection Results on IITR dataset 28
4.6 Existing Problems in text Detection 29
4.7 HOG features's size calculation in an image 30
4.8 (8,1) neighborhood in LBP features calculation 31

4.9 Bar chart showing precision, recall, and F-score of using different features with SVM classifier 32
4.10 Bar chart showing precision, recall, and F-score of using different features with Naive Bayes classifier 32
4.11 Bar chart showing precision, recall, and F-score of using different features with KNN classifier 33
4.12 Bar Chart representation of precision, recall and F-score of different classifiers 35
4.13 An Architecture of Regular Neural Network 36
4.14 An Architecture of Convolutional Neural Network 37
4.15 Maxpooling process on an image 38
4.16 Proposed CNN model 38

CHAPTER 1

INTRODUCTION

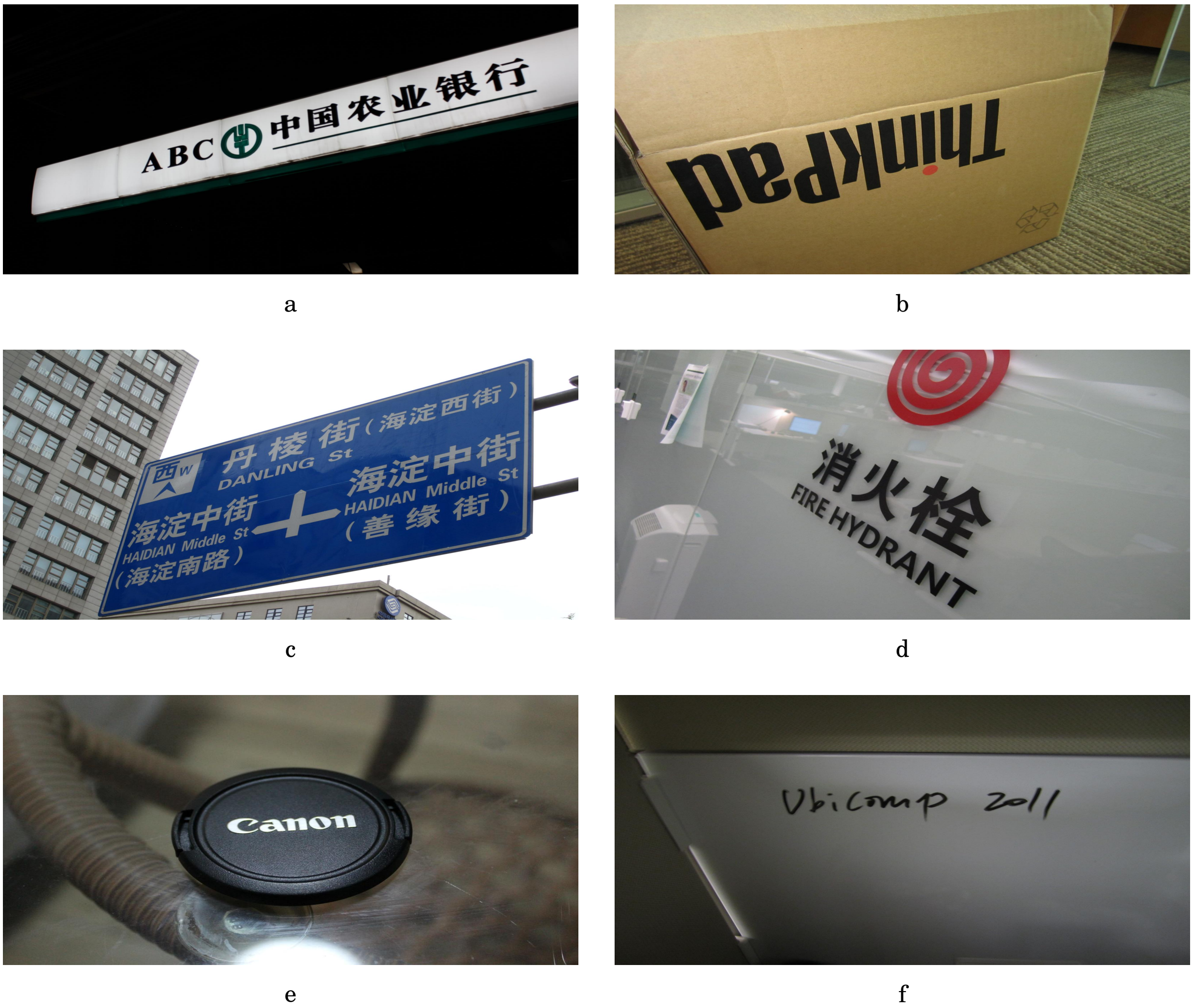
Text is one of the most effectively conveying means of communications, and text can be embedded into documents or into scenes as a way of conveying information. Recognized text can be useful for different purposes such as,

1. A text recognition system can function as a vision for wearable and can be used for translating languages.
2. Recognized text can be used in a license/container plate recognition system
3. A text recognition system can be used for the automatic structuring of videos or images.
4. Recognized text can be used to analyze the content involved in the video or images.
5. Text information on each part in an industry can be used for their identification and other purposes in industry.

The problems of text detection, extraction, and recognition in images and videos have gained large attention in recent years. In complete process of text recognition, text detection is the primary and fundamental step. Thus, we need to focus on text detection which in turn increases the accuracy of text recognition system. Different text properties make it more difficult to detect text. These text properties are explained below.
Geometry Alignment: In the natural scene, text may have various distortions. Text can pursue geometric distortions and can be aligned in any direction.
Size: Text size can vary within an image or video frame.
Inter-character distance: Characters in a line generally have the same distance in between.
Color: The characters in a text can have same colors. But multiple color strings may be present.

Edge: Most scene text and caption are designed to be readable, therefor the boundaries of text and background have strong edges.

Figure 1 depicts the variations in the text in video frames and images. Figure 1.1 contains images from MSRA Text Detection 500 (MSRA-TD500) dataset.Figure 1.2 contains frames of videos from International Conference on Document Analysis and Recognition (ICDAR) 2015 dataset. The text within these images and video frames clearly varies in size, texture, intensity, alignment etc.



a

b

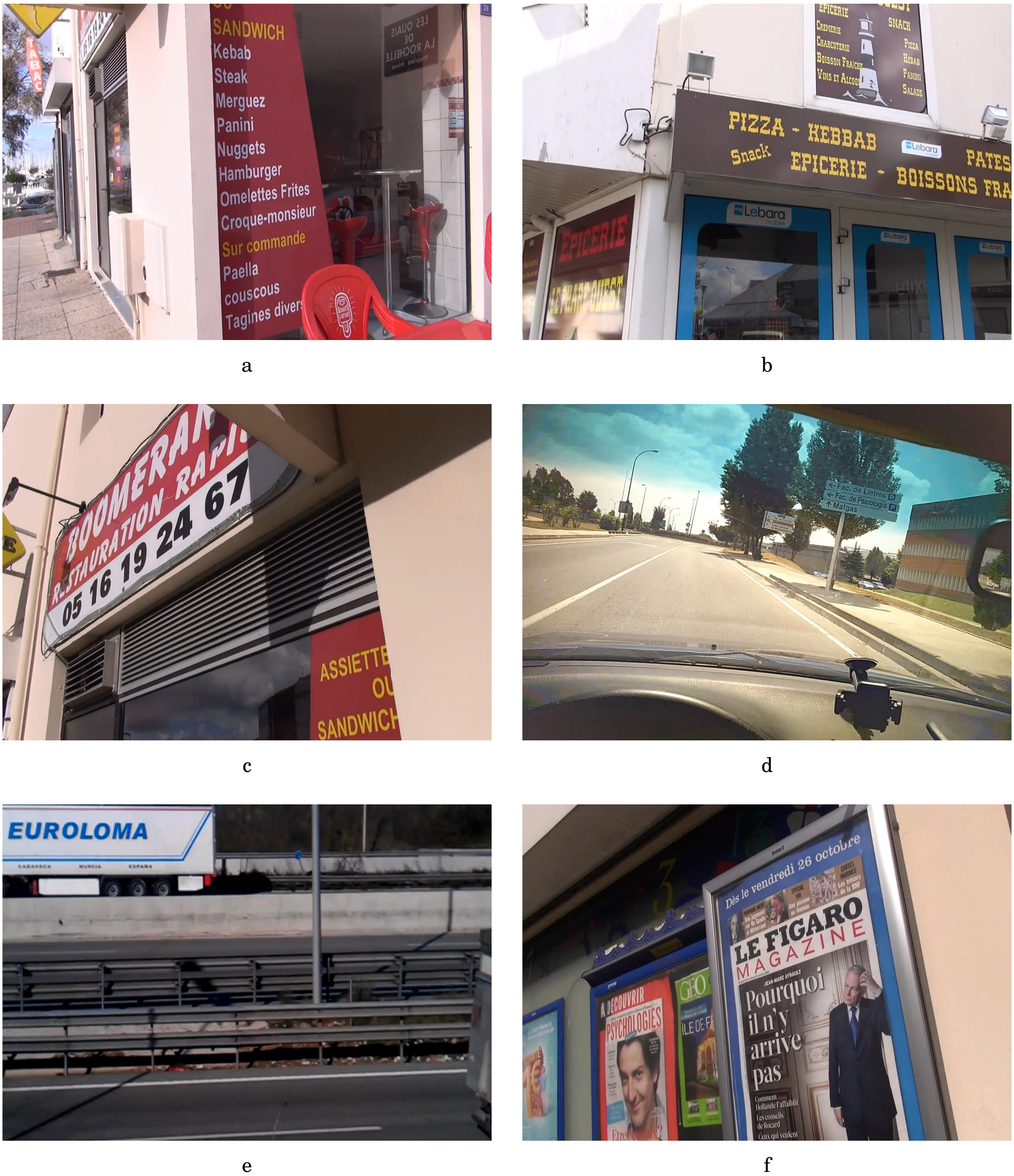
c

d

e

f

Figure 1.1: Figure showing variations of text in images from MSRA-TD500 dataset



a

b

c

d

e

f

Figure 1.2: Figure showing variations of text in video frames from ICDAR dataset

Although the text extraction and recognition give rise to manifold applications, the foundational object is to detect the text in images or video , and it is important to overall system performance. Various approaches have been proposed for text detection such as,[2],[3],[7],[8],[9],[10],[11] ,[15] etc..

1.1 Fundamental Sub Problems in Text Information Extraction

Text Information Extraction (TIE) system is divided into the following sub-problems: (a) text detection (b) tracking of text in continuous frame, (c) extraction and enhancement of text, and (d) text recognition (OCR)

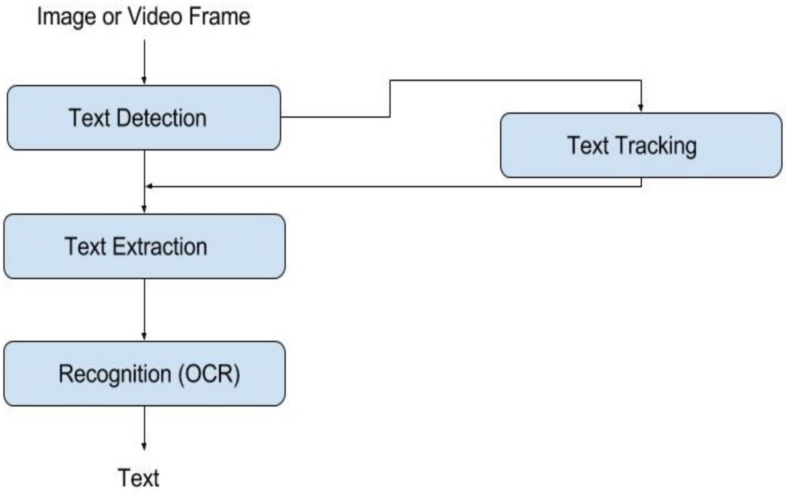


Figure 1.3: Architecture of the TIE System. [23]

1.1.1 Text Detection

Most of the TIE system assume that text is present in image or video. This assumption is very frequent for scanned images. However, for a natural scene, this cannot be true. They may or may not contain text. Generally, the text shows a higher intensity in respect to the background. If a maximum number of pixels are present in a frame with intensity lighter than a threshold and show a color variation with their neighbors, this frame is regarded as a text frame. This method is very fast and extremely simple. But for simplicity, we assume that every frame has text and thus focuses on text detection. Most of the text detection methods are divided into two types on the basis of features utilized: region-based and texture-based.

- ***Region-based methods***: Region-based methods utilizes the properties of the gray scale or color in a text region in comparison with the background of the text. These methods can be further divided into edge-based and connected component (CC)-based methods.

  Edge-based methods: This type of method use high contrast between the background and the text. The text boundary edges are founded and then merged. Further, non-text region is filtered out using several heuristics.

CC-based methods: CC-based methods identify all the region in an image by grouping small components into larger one. Text boundaries are marked using geometrical analysis. Binary segmentation is used in the CC-based method. However, video document may have various gray level objects, therefore, binary segmentation is inappropriate. Also in this approach, there are problems associated with text alignment and color of the text.

- ***Texture-based methods***: Researches show that, text in the images or videos have different textural properties than the background and this property is used in Texture-based methods. In this, textural properties of a text region are identified using Gabor filters, Wavelet, FFT, spatial variance, etc. There are also some other approaches of localization of text such as, Binarization, which utilizes adaptive, local or global thresh holding. These methods are widely on the images having white background with black characters on them, and thus, segmentation can be done easily. This approach is used in application such as courtesy amount on checks, address location on postal mail, etc.

1.1.2 Tracking, extraction, and enhancement

Tracking in the video has not been studied extensively despite its various uses. Same is the case with extraction and enhancement either. Temporal change in frame sequence is needed for speedy system performance. Text detection results can be tracked using text tracking. Text tracking can also be used in recovering original frame where the text is hindered in different frames.

1.1.3 Text recognition

Detected text can be fed into a recognition system like Optical Character Recognition (OCR). OCR system results into recognized text which can be useful for various purposes.

1.2 Enhancement of Text Using Super Resolution

Text block or a particular frame containing text can be produced in high resolution using super resolution technique. For this, we first need to know about super resolution,

1.2.1 What is Super Resolution?

Super resolution is a process of obtaining high resolution image or image sequence or frame from a series of low resolution (noisy) images or frames. With this way, super resolution constructs a high resolution image of the initial image or frame with the help of observed images or frames having lower resolution. Super resolution algorithms utilizes the existing images or frames and thus, it reduces the cost of overall process of constructing high resolution images. In videos, a common reference frame is used to map obtained frames. This process of mapping a referenced frame with obtained frames is called registration. Then, the super resolution is applied on the registered image. The successful super resolution mainly dependent upon the two things, first precise registration process of the images possessing low resolution and second is forming a proper observation model.

1.2.2 Image Registration

The obtained low resolution images have a distinct view of the same scene. Thus, we can get a high resolution image by mapping appropriate pixels in the images having low resolution. This process of mapping pixels or points of images of low resolution or low quality is called registration. Registration process is not effective until there is low resolution images with sub-pixel shift. If obtained low resolution images have pixel shift at unit level, then any additional information required for the generation of the high resolution image will not be available, as the pixels of different lower resolution images will be overlapping . If observed images are shifted at the level of the sub-pixel, then there will be extra information and high resolution image can be created using this extra information. Figure 1.4a presents the representation of pixel on a 2-D grid of first lower

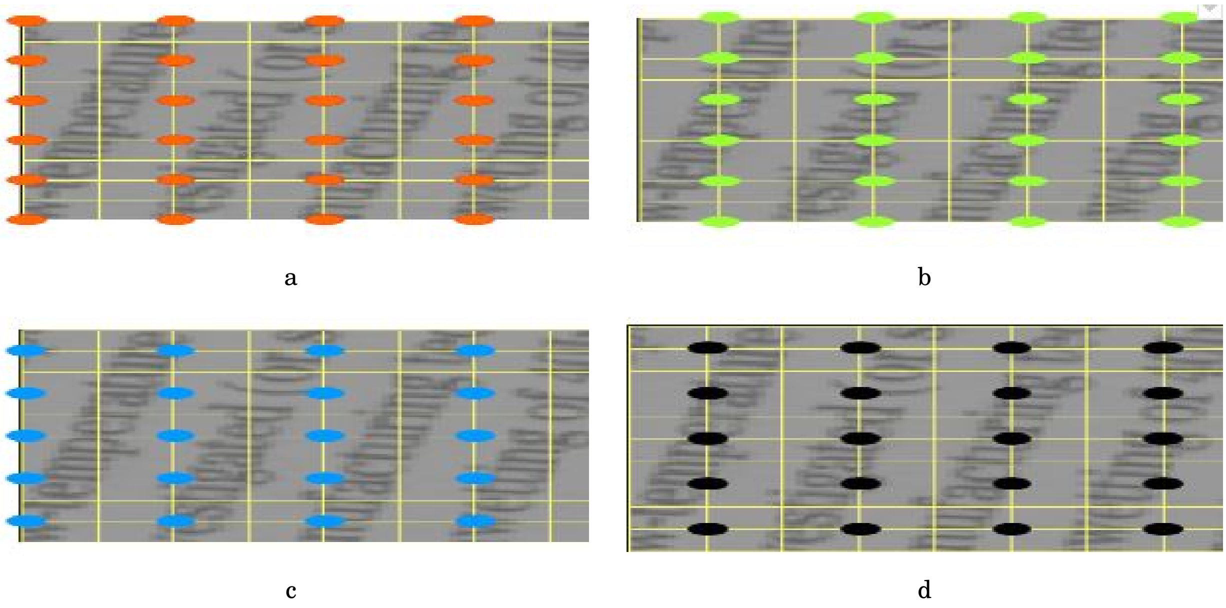


Figure 1.4: 2-D representation of pixels in images taken at sub pixel level. Here, (a) represents low resolution image (Reference Image), (b) second low resolution image, (c) third low resolution image, and (d) fourth low resolution image. (taken from http://www.cse.buffalo.edu/)

resolution reference image. Figure 1.4b shows the second picture which is obtained by moving the camera slightly to the right. Figure 1.4c presents the third image obtained by moving the camera slightly downward. And figure 1.4d shows fourth image generated by shifting the camera slightly to the right and downward. Further, registering images which are described, we get a high resolution image, that is shown in figure 1.5. Here in super resolution image, each block contains more no of pixels as compared to low resolution images. Thus, contained information is more in high resolution image. Basically,image registration process is divided into two types, i.e. photometric image registration, and geometric image registration.

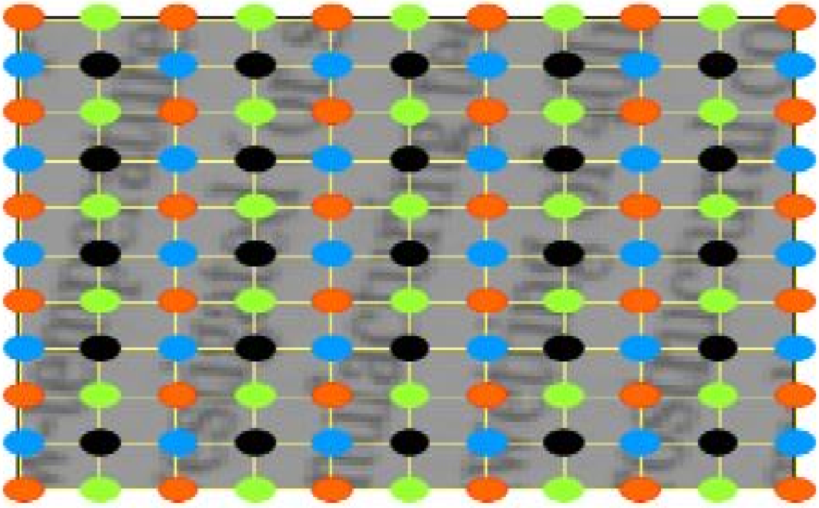


Figure 1.5: Obtained high resolution image by registering four lower resolution images at a sub-pixel level.(taken from http://www.cse.buffalo.edu/)

Photometric Image Registration Low-resolution images may possess photometric changes at different level. Photometric registration proves to be helpful in recording the changes of contrast,color intensity, and brightness. Further, this information is used in enhancing the registration of the low-resolution images.

Geometric Image Registration This type of registration considers the points or pixels geometry in distinct low resolution images. The images can be clicked from a camera stationed at different locations and also panning and zooming functionality of the camera can be incorporated. Suppose there are z and z' points in two different low resolution images of a same low resolution image which represent a point X in the original image or scene. In geometric registration points from different images are projected on a planar surface and then registered.

1.2.3 Algorithms for Super Resolution

1. Non-uniform Interpolation

This is one of the techniques of super resolution in which high resolution image is reconstructed from images with low resolution. This technique consists of three stages

1. registration of the images with low resolution.
2. using non-uniform interpolation.
3. image de-blurring applied on the produced image.

Figure 1.6 shows the non-uniform interpolation process. In this process, first motion information is estimated between observed low-resolution images. And this motion information is used in performing registration between observed images with low-resolution. On the registered image, an interpolation which is not uniform is applied using using direct image reconstruction or iterative image reconstruction. This will generate an image with high resolution. Then noise removal method are applied and noise introduced during non-uniform interpolation process is removed.

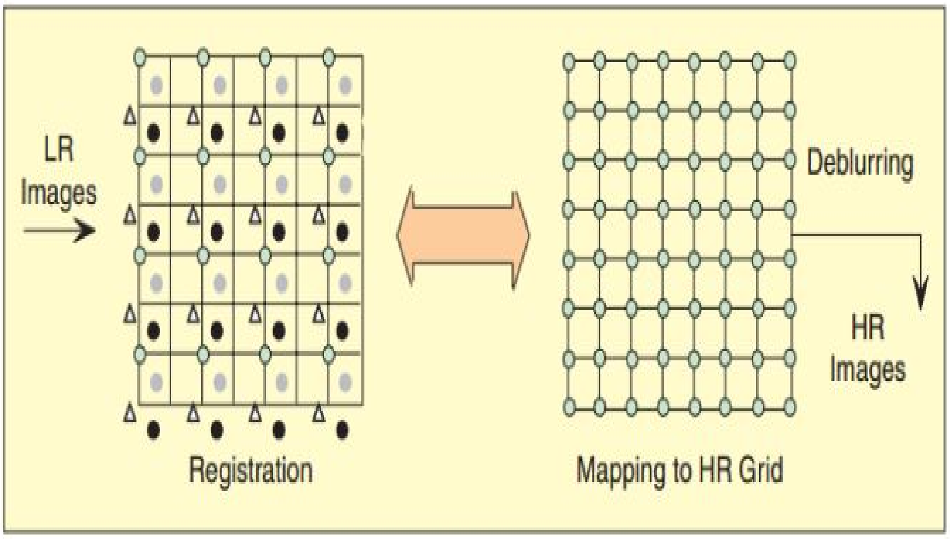


Figure 1.6: Details of non-uniform interpolation. [22]

2. Iterative Back Projection (IBP)

Irani and Peleg [26] developed IBP. This method uses image blur to simulates the LR images. Then the difference between observed image and simulated LR image is calculated. This differences (error) is used to generate a high resolution image by back projecting it. IBP have some edge over other method of super resolution, that it is very easy to apply and understand. However there are some disadvantage also i.e. an unique solution can not be guaranteed by this method and addition to this, hBP (back projection kernel) parameter must be chosen for this method. hBP parameter is used to calculate the error contribution. And also, final high resolution image is dependent on hBP parameter. As super resolution problem is ill-posed in nature .i.e. if an image with high resolution is broken down into a number of low resolution images, then no unique solution is guaranteed. In other words, we can say that many high resolution images can produce an unique low resolution image. "Due to this ill-posed nature of the super resolution problem, there is difficulty in choosing hBP" [22].

3. Papoulis-Gerchberg Algorithm

Papoulis-Gerchberg Algorithm [16][17] considers two things,

1. It considers a high resolution grid which has some known pixel values.
2. High-resolution image contains zero components with high frequency.

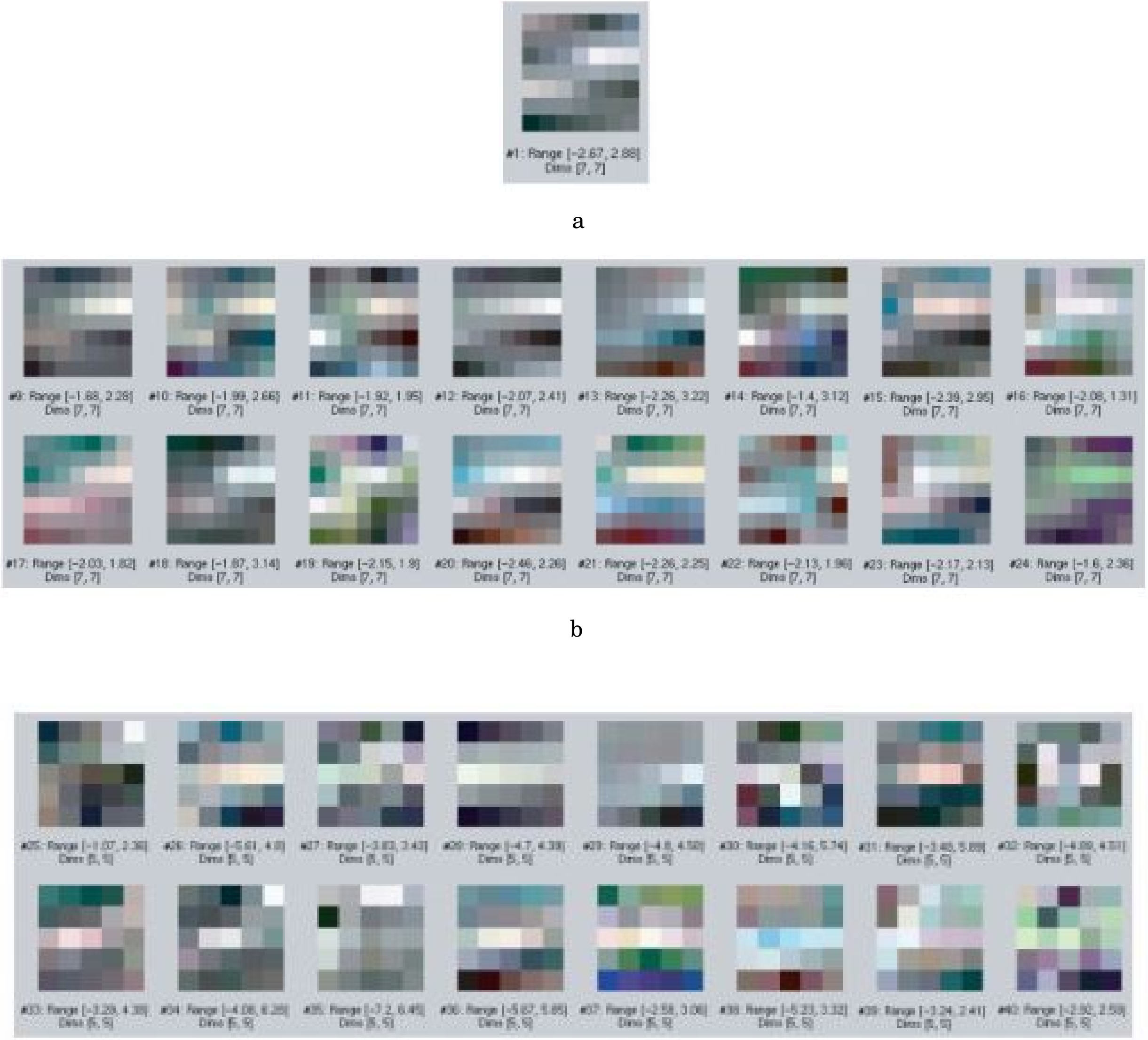
Papoulis-Gerchberg Algorithm is an iterative process and iterates over below steps,

1. Use a low-pass filter to remove high frequency components and make a high resolution grid.
2. Insert the known pixel values from low-resolution image to high resolution image on a position rounded to nearest integer location.
3. If the process does not converge then iterate from the first step.

The unknown values are interpolated by setting high-frequency components to zero and thus, low-frequency component aliasing is corrected. Also, the value of the some of the high frequency component is predicted by inserting the known pixel values.

4. Example Based Super Resolution

Researchers noticed that patches recur within and across the scale of an image. And this property is utilized in example-based super resolution to generate a high resolution image from low resolution image. In Example-based super resolution, a database of patches of high and low resolution pairs is created generally with a scale factor of two. This algorithm learns a correspondence between these patches and then a high resolution image version is obtained from a new low resolution image. Nearest neighbor search is used to select high resolution patches from already selected high resolution patches and low resolution patches from the database. By repeating this process a higher resolution factor can be obtained. Although, in the example-based super resolution we can have a high resolution image from a single low resolution image but true high resolution details can not be guaranteed.



a

b

c

Figure 1.7: Retrieved patches from database for an input low resolution patch. Here, (a) represents an input patch from low resolution image, (b) represents closest image patches from database, and (c) represents corresponding high-resolution patches from database. [27]

1.3 Dataset Description

The dataset used for the experiments purpose is ICDAR 2015 dataset. It consist of 49 videos, 25 of which are given for training purpose having 13450 frames in total and 24 for testing purpose having 14374 frames in total. The text presented in videos is multi oriented. furthermore, it is multilingual dataset as it contains text of Spanish, French, English, and Japanese. Table 1.2 describes various properties of the videos present in ICDAR 2015 dataset.

Another dataset used for text detection is IITR dataset. This dataset consists of

Table 1.1: Detailed description of videos from ICDAR 2015

Type	Size Range (MB)	Time Range (second)	Aspect Ratio Range (width x height)	Frame Rate (per second)
Training Videos	2.33 - 35.3	10 - 40	720 x 480 - 1280 x 960	24 and 30
Testing Videos	1.16 - 33.8	5 - 38	640 x 480 - 1280 x 960	24 and 30

multiple videos of Indian streets. Also, there are videos of multiple indian languages like Hindi, Malyalam and Bengali. In some videos, English and Urdu words are also present. Our proposed method shows efficient result on this dataset. Table 1.3 describes various properties of the videos present in ICDAR 2015 dataset. Aspect ratio of videos in whole data set is either 1280 x 720 or 640 x 480.

Table 1.2: Detailed description of videos from IITR videos dataset

Set	Number of videos	Size Range (MB)	Time Range (second)	Frame Rate (per second)	Languages
Set 1	5	0.6 - 15.2	2 - 125	30	Hindi and English
Set 2	5	0.6 - 4	2 - 21	30	Bengali, Hindi and English
Set 3	4	2 - 3	5 - 8	30	Malyalam and English

1.4 Overview of the problem

Although most of the existing approaches have given good performance in text detection , there are certain problems which need to be addressed such as sometimes we have smaller text which is difficult to detect.In images or videos distorted text may be present or we may have low-resolution images or videos,which degrades the performance. To overcome these problems we have proposed a method which uses super resolution technique and MSERs features. In our proposed method we first find key frames from videos and apply Papoulis-Gerchberg[16][17] algorithm (a super resolution technique) for enhancing the resolution. For further increasing the accuracy of text detection we map the detected region from low-resolution image and high-resolution image. At this level, we have detected text regions which may contain NonText region also. Therefore, there is a need of classifying the detected regions as Text and NonText. For this purpose, we train a classifier on HOG(histogram of Oriented Gradient) and LBP(local binary patterns) features. In experiment section, we present a comparative analysis of different classification technique like Support vector machine(SVM), K-nearest neighbors (KNN) and Naive Bayes on our dataset. We also use deep learning's CNN to classify text and non text.

CHAPTER 2

RELATED WORK

2.1 Some Basic method of text detection in images and videos

Various methods have been proposed with number of techniques for text detection like, Smith et al. in [1], proposed a text detection method in video frames. They considered text region as a "horizontal rectangular structure of clustered sharp edges" [1] due to difference in intensity of characters with its background. and used this property to extract textual information from video frames. This method is scale dependent, i.e., it detects text of certain size range. A K Jain et al. in [2], proposed a method of text detection for images and videos. This method incorporate multivalued image processing. Multivalued images are having pixel values in the range of {0,1,......,M-1}, where M is an integer and $M > 1$ [2]. These multivalued images are decomposed into multiple foregrounds and background-complementary foregrounds images. Then connected component analysis is applied and text is detected. This method is applied on advertisement images, color images, web images and video frames. . Figure 2.1 shows the overview of method [2] and Table 1.1 shows the accuracy of this method on different data.

In [9], Shivakumara et al. performed K-means clustering to obtain connected components in the Fourier-Laplacian domain. And finally, edge density and text straightness are used to eliminate false positive.In [4], D Chen et al., proposed a method for text detection and recognition in images and video frames. For detection of text, this approach vertical and horizontal edge are first detected by a Canny filter. The text regions are those which are covered by both vertical and horizontal edge dilation. Different dilation operator are used for vertical and horizontal edge. This approach of text detection is invariant to intensity. Further for removing false detection, top and bottom "baselines of horizontally

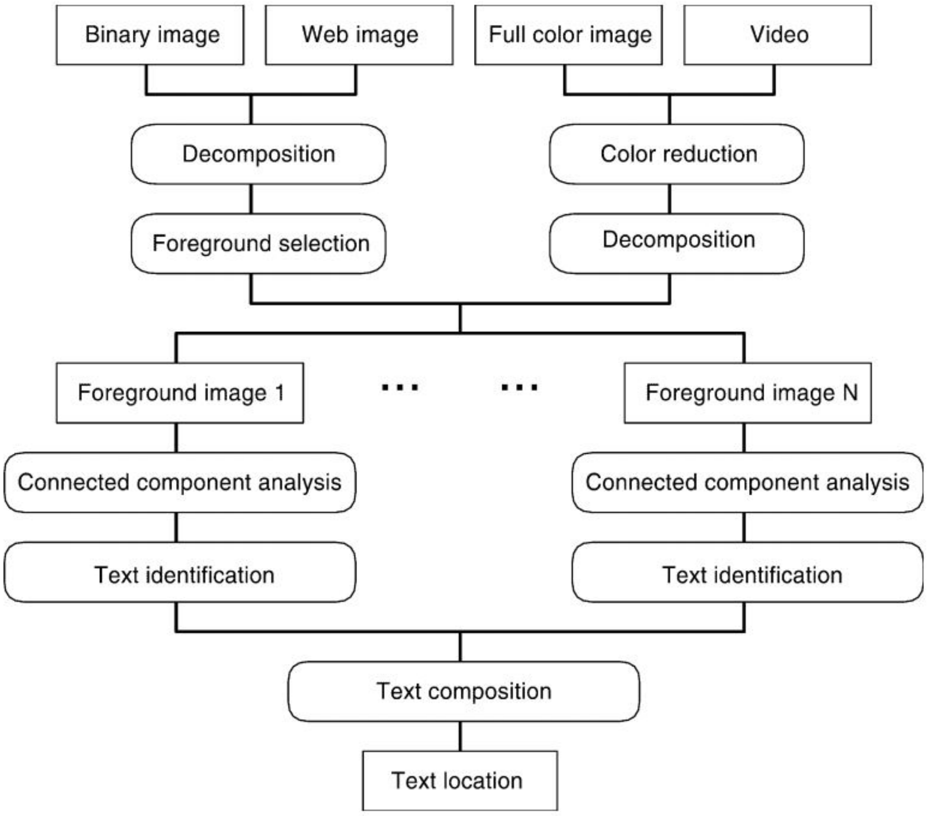


Figure 2.1: Overview of method proposed by A K Jain et al. [2]

Table 2.1: Accuracy on different dataset of method proposed by A K Jain et al.

Text Carrier	No. of Test images	Typical Size	Accuracy
Advertisement	26	548 x 769	99.2
Web Images	54	385 x 234	97.6
Color Image	30	769 x 537	72.0
Video Frame	6952	160 x 120	94.7

aligned text strings are detected" [4]. False detection normally does not contain any well defined baseline.

2.2 Text detection in images and videos using neural networks

In [3], K Jung proposed a method which utilizes the neural network for text detection in color images. Method extracts texture information on different color bands and then this information is combined using the neural network and used for text detection. In [13], R Lienhart et al., used a multilayer feed-forward network to detect text lines. They also exploited temporal redundancy of text in videos for enhancing text detection. H Li et al. in [12], used a hybrid wavelet/neural network based method to detect text regions in videos. They also used temporal redundancy for text tracking. In text tracking, initial position of text is find using sum of squared difference and final position is detected by contour-based technique. To detect different text size they used image pyramid with each level halved to previous level resolution. The extracted text from different level are rescaled to original size. Figure 2.2 shows a method for text detection and tracking in digital video as depicted in [12]. And figure 2.3 depicts the architecture of whole process proposed by [12].

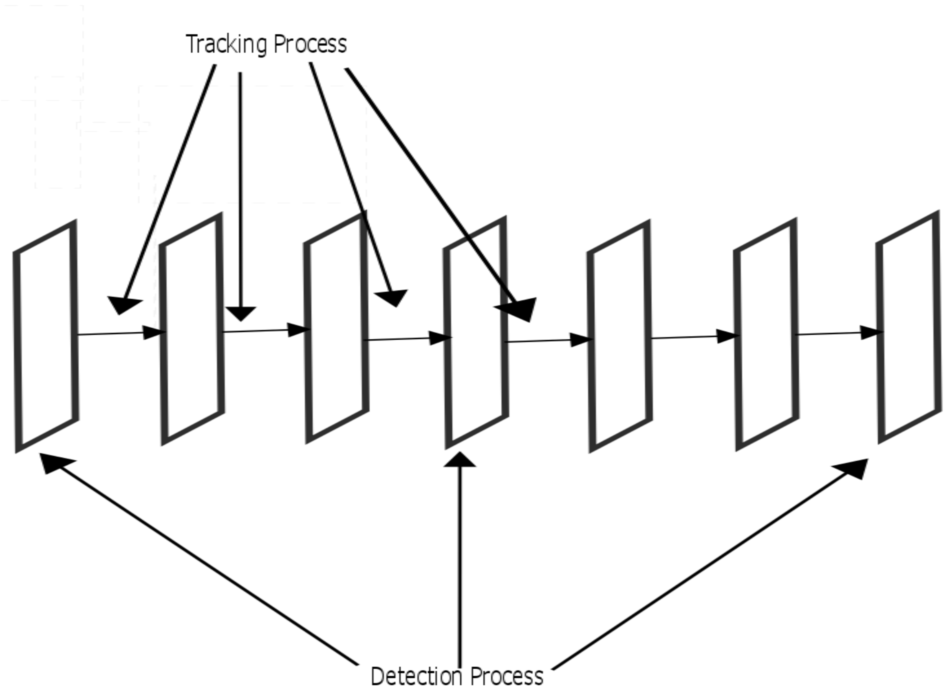


Figure 2.2: Overview of text detection scheme prorposed by H Li et al. [12]

2.3 Text detection in images and videos using MSERs

In [11], C Shi et al. used MSERs descriptors for text detection. They considered text detection problem as segmentation problem of text and non-text regions. In this method, first

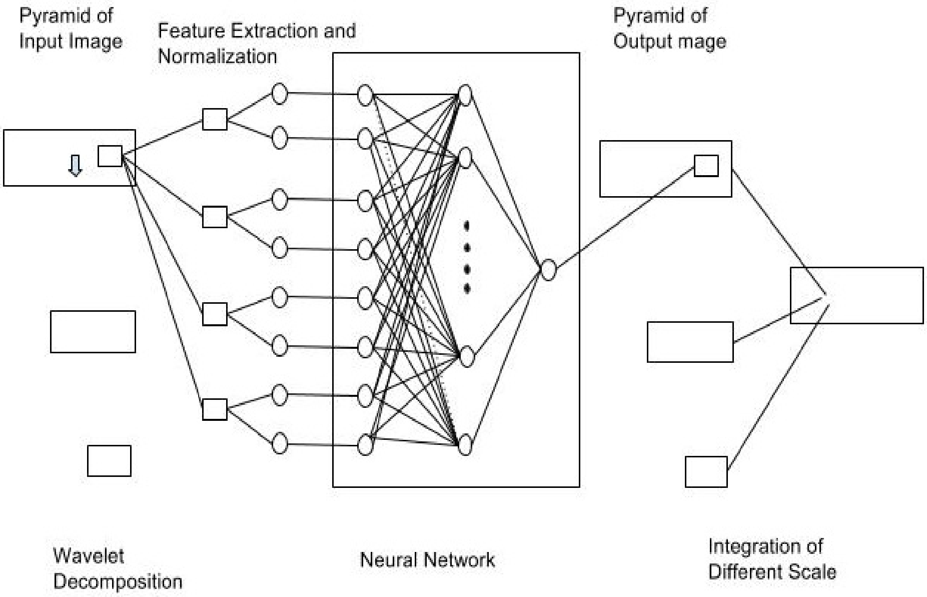


Figure 2.3: architecture of whole process proposed by H Li et al. [12]

MSERs are detected on the original image and then an undirected graph is constructed upon these MSERs. Using this graph model, the graph nodes (MSERs) are specified as text and non-text. Figure 2.4 depicts the flowchart of algorithm proposed by [11]. In [10],

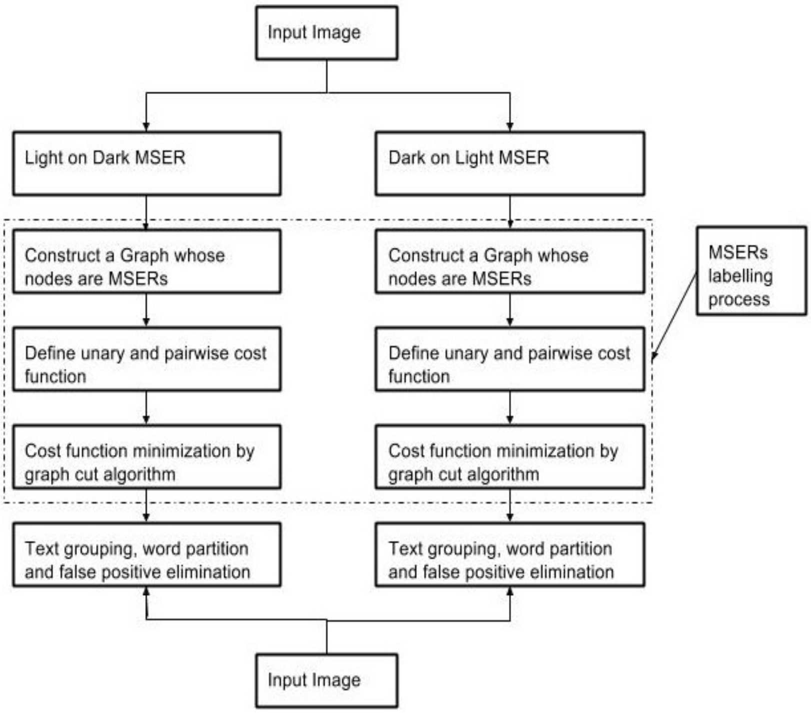

Input Image
Light on Dark MSER
Dark on Light MSER
Construct a Graph whose nodes are MSERs
Construct a Graph whose nodes are MSERs
MSERs labelling process
Define unary and pairwise cost function
Define unary and pairwise cost function
Cost function minimization by graph cut algorithm
Cost function minimization by graph cut algorithm
Text grouping, word partition and false positive elimination
Text grouping, word partition and false positive elimination
Input Image

Figure 2.4: Flowchart of algorithm proposed by C Shi et al. [11]

H Chen et al. combined Canny edges and MSERs to obtain edge enhanced MSERs. And then non text regions are removed using stroke width and geometric information. This process is illustrated in Figure 2.5.

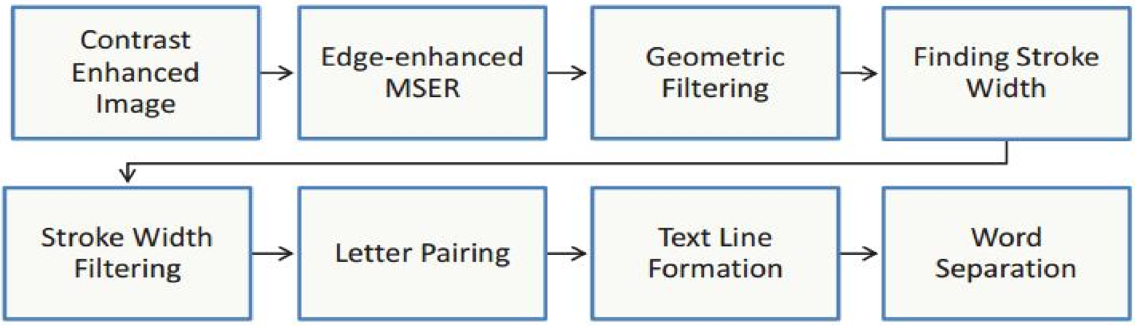


Figure 2.5: Illustration of process proposed by H Chen et al. [10]

In [14], Y Li and H Lu, used MSERs with enhanced contrast for text detection. MSERs are affected by image blur, the method overcome this problem of MSERs. Further, geometric filtering is used to remove non text regions. W Huang et al. in [5], also used MSERs for text detection problem. They use CNN to separate texts from the background. XC Yin et al. in [6] detects candidate text regions using MSERs. They removed repeating components and grouped text candidate.They further used an AdaBoost classifier to classify text and non-text regions.

Chapter 3

Proposed Work

This chapter consist of detailed discussion about the proposed method. Outline of proposed method is represented by the Figure 3.1. This method is mainly divided into 5 steps as,

1. Extraction of Key Frames from Videos.
2. Applying Super Resolution.
3. Detect MSERs regions.
4. Mapping HR image and LR Image.
5. Using a classifier to separate text and non text regions.

3.1 Extraction of Key Frames from Videos

A keyframe is a frame which can represent a particular section or whole video in one or few frames. In a video consecutive frames contains the similar details or features. So we need keyframes, to detect MSER features. There are different standard methods for key frame extraction such as, key frame selection by motion analysis [19] , use of unsupervised clustering [21], key frame extraction by shot selection [20] etc. These methods provide a unique key frames for a section of a video.

As we are using super resolution method for generating High Resolution Image with the help of key frames, if there are more number of key frames there will be more time complexity. Therefore we should have such number of key frames so that time complexity could be minimize and most importantly we should get every text possible in the videos.

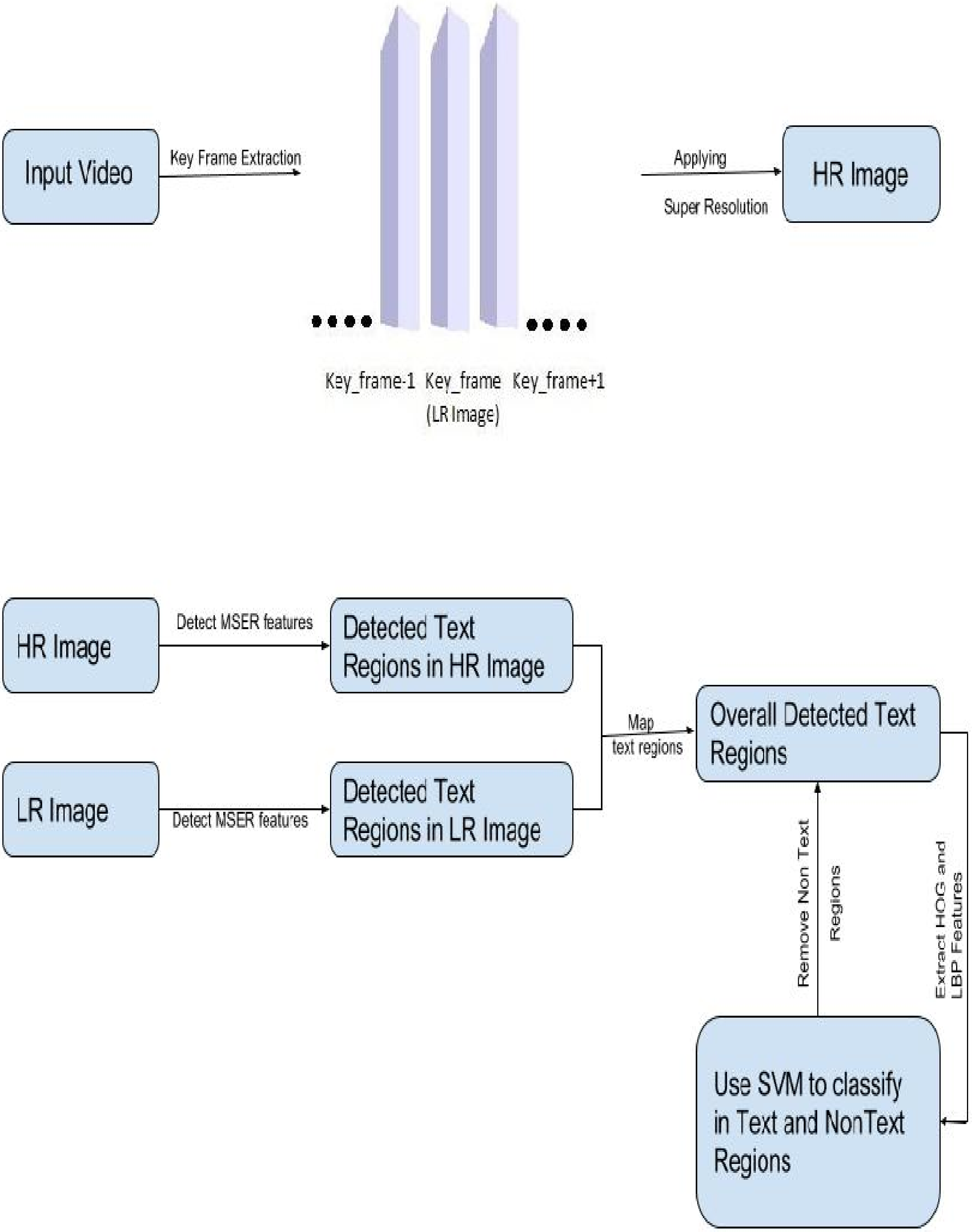


Figure 3.1: Outline of the proposed method

Therefore we are using histogram difference comparison method for calculating key frames. The methods based on histogram difference comparison are less sensitive to since they do not entertain spatial changes in a frame. Object motion caused by moving vehicle, person etc. and it can affect the text detection, if key frame selection method removes a frame having more number of text than the selected frames. Smith et al., in [1], used comparative histogram difference approach for scene segmentation. A Nagasaka et al. in [?] and H Zhang et al. in [?] also used histogram comparison methods for scene segmentation. In this method we are calculating absolute histogram difference of consecutive frames and then setting threshold as mean of these calculated values. Then absolute histogram difference of frames is compared with the threshold value. If it is having low value than the threshold, it is discarded otherwise considered as key frame. By setting mean of absolute histogram difference as threshold this method provide an adequate number of frames.

```
input  : A video
output : Key frames
while not the end of the video do
    if selected frame is not last then
        read current frame and next frame;
        calculate absolute histogram difference;
    end
end
set threshold=mean of absolute histogram difference;
while not the end of the video do
    if selected frame is not last then
        read current frame and next frame;
        diff=calculate absolute histogram difference;
        if diff > threshold then
            select frame as key frame;
        end
        else
            discard the frame;
        end
    end
end
```

Algorithm 1: Key frame selection using histogram difference

3.2 Applying Super Resolution

Super resolution is a process of generating high resolution images from low resolution images.The high resolution image has more details as compared to low resolution images. There are may super resolution algorithms are available [22], in our case, we are using

Papoulis-Gerchberg[16][17] algorithm for super resolution with an interpolation factor of two (we are getting good HR image quality and less time complexity as compared to greater factor). For successful super resolution method requires good motion estimation (required in low resolution image registration).For this purpose, we are also using previous and next frames of each key frame. These three frames are used to estimate motion and Papoulis-Gerchberg algorithm is applied to obtain the high-resolution image (HR Image). This algorithm is explained in section 1.3.3.3 of chapter 1.

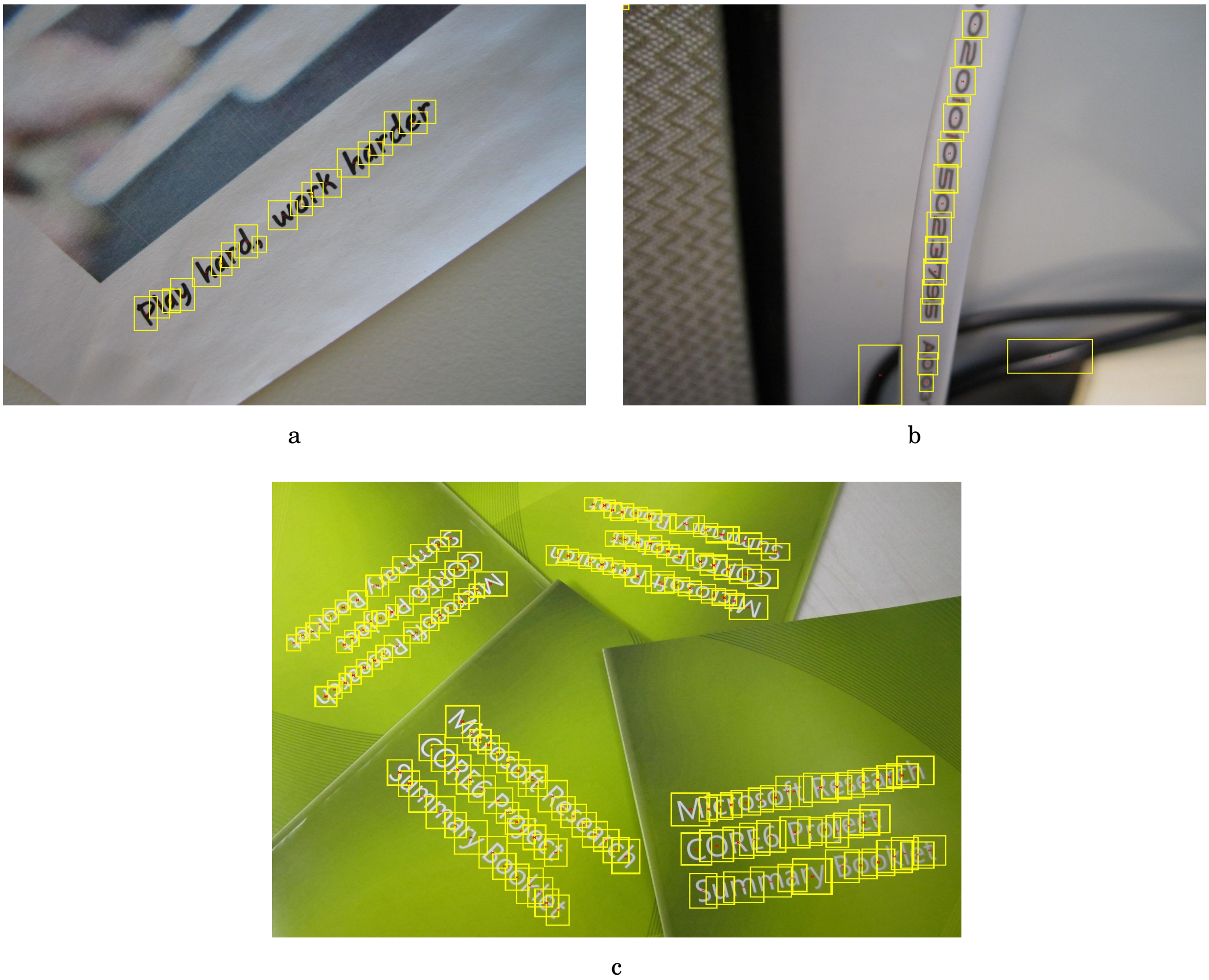
3.3 Detect MSERs regions

Maximally stable extremal regions (MSERs) are used for blob detection in images. Matas et al. [18] proposed this technique to find similarity between different image parts. Since text generally has uniform intensity or color and distinct contrast relative to its background, MSERs is a instinctive choice for the detection of text. Some of the properties of MSERs are listed below,

1. The computation of MSERs is very efficient. All extremal region in a image or frame can be retrieved in $O(n)$, where n represents total number of pixels in an image or frame.
2. MSERs detector detects the uniform intensity regions in an image or a frame. And generally, text in an image or a video frame is considered to be having uniform intensity.
3. MSERs detector shows robustness against viewpoint.
4. MSERs detector is also robust to scale.
5. As uniform intensity regions are detected, it is language independent. It can detect text of any language.

As text possess contrast to its background and uniform intensity or color ,MSERs is a good choice for text detection. We detect MSERs regions on HR image and get probable regions of text. For increasing accuracy in text detection, we also detect MSERs features on corresponding key frame (considered as LR Image). Figure 3.2 shows that MSERs detector detects text of any orientation i.e. robust to viewpoint. In figure 3.2, all images are from MSRA-TD500 dataset.

Figure 3.3 shows that MSERs detector detects text independent of languages. We have applied MSERs detector on images containing various languages and results are shown in figure 3.3. Figure 3.3(a) and 3.3(b) are form MSRA-TD500 dataset, while image 3.3(c) is taken from *http://techwelkin.com* .



a

b

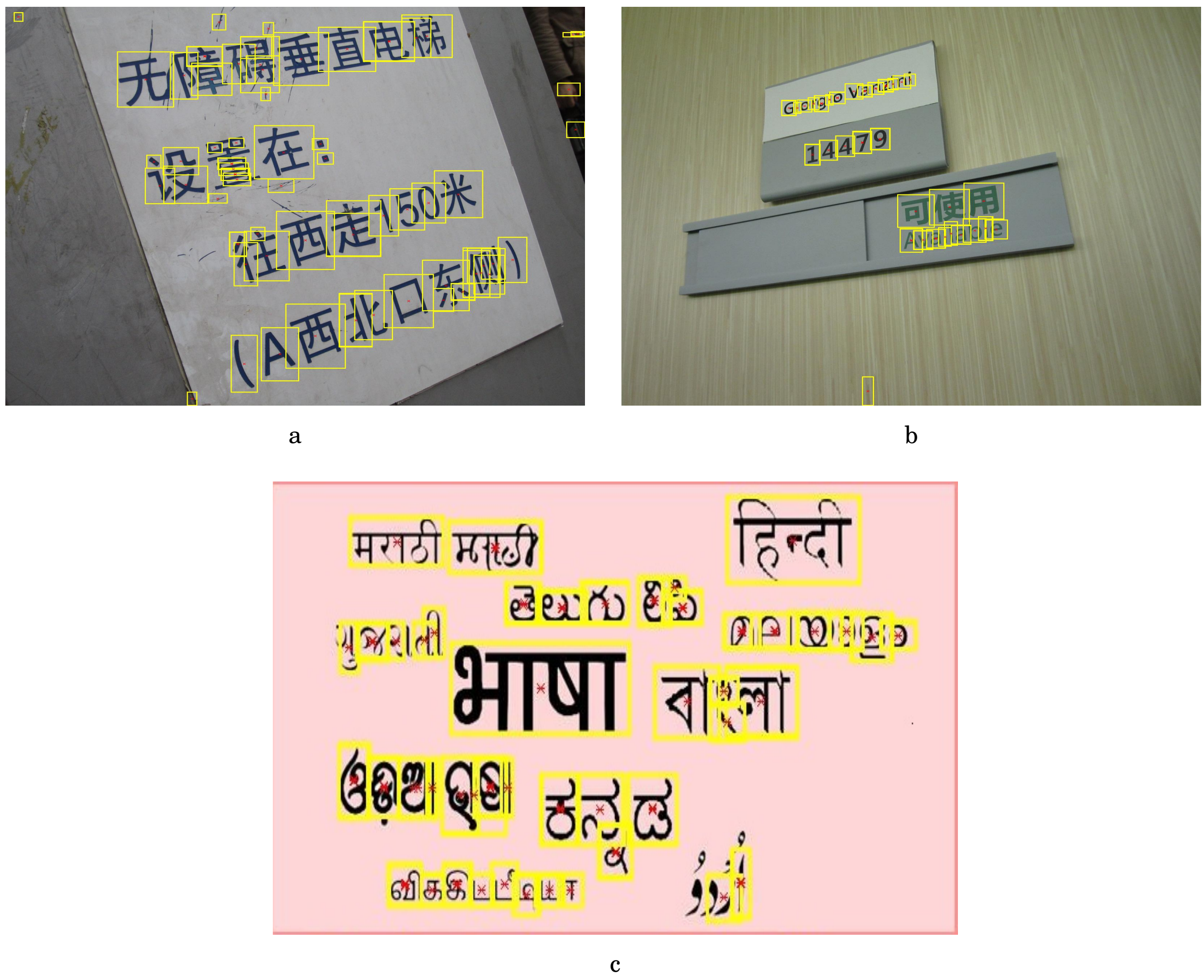
c

Figure 3.2: Figure showing robustness of MSERs text detector against view point

3.4 Mapping HR image and LR Image

There is a possibility that the generated high resolution may miss text regions in detection which are detected in the LR image due to lighting changes. Therefore, to increase accuracy of text detection we detect MSERs regions both in HR image and LR image. Further to this, detected regions need to be mapped onto LR image so that detection result improves. In figure 3.4, arrow points to detected regions in LR image which are missed in HR image.

As we have increased the number of pixels in HR image in both the direction by a factor of two. HR image is two times greater than the LR image in both directions. We map the text regions obtained from HR Image onto the LR Image as shown in Figure 3.5.For every obtained region in HR Image, we half the x-coordinates and y-coordinates and thus new coordinates are obtained,which mapped onto the LR Image.Now LR Image contains over all detected regions. These are probable text regions but may also be nontext regions.



a

b

c

Figure 3.3: Figure showing robustness of MSERs text detector against view point

3.5 Using a classifier to separate text and non text regions

As overall detected regions may contain nontext regions. Therefore there is a need of removing Non Text regions from detected text regions, which in turn increases accuracy. This step is further divided into two sub-steps.

3.5.1 Feature Extraction

This step is performed by extracting the Histogram of Oriented Gradients(HOG) and Local Binary Patterns (LBP) features of the each text regions. We used 2x2 cell size for hog feature extraction. As block size represents the number of cell in a block. If we take larger block size then local illumination changes may affect the result. Changes in illumination of HOG features can be suppressed by small block size. Since HOG performs on the local cells,

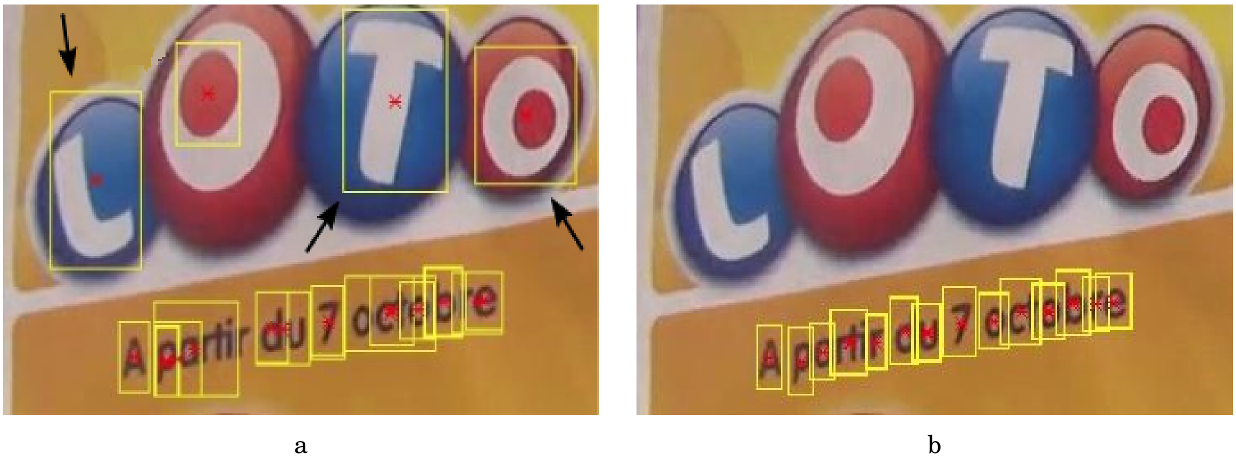


Figure 3.4: Figure showing robustness of MSERs text detector against view point. In this Figure, (a) shows detected regions in LR image, and (b) shows Missed regions in HR image.

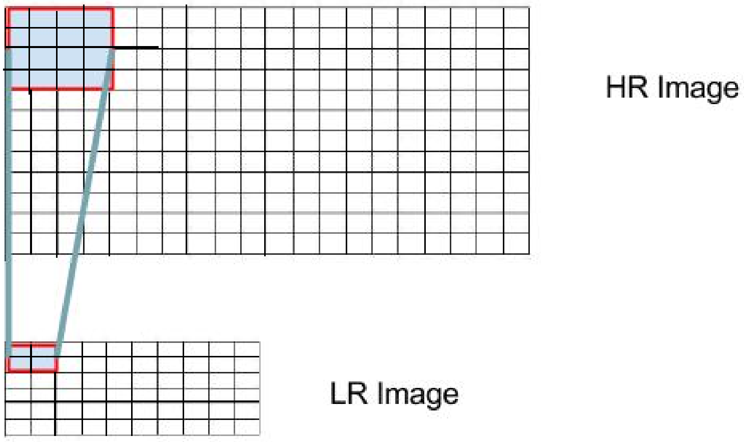


Figure 3.5: Mapping Detected Regions from HR Image onto LR Image.

1. It is invariant to geometric transformation except rotation and translation. In geometric transformation, scene points from different dimensions are mapped onto an imaginary plane and it may be of the type such as translation, scaling, rotation, and skewing.

2. it is invariant to photometric transformation. The photometric transformation determines the intensity at different image points.

LBP features represent local texture information. It has following properties,

1. LBP operator is robust to monotonic changes in gray scale. These changes may be caused by illumination variation.

2. LBP operator can prove robustness against rotation.

3. To calculate LBPs, very few operations are performed, implying its computational simplicity.

For applying HOG each image should be of the same size, therefore, we first resize is each text regions to 20 x 20 pixels. Further to this, HOG and LBP features are extracted from images of size 20 x 20. This step produces a vector of data points which carry much information about the detected region. These data points are the combined HOG and LBP features.

3.5.2 Classification

Our text detection result gives many false positive, therefore there is a need of removing nontext regions from the final result. For comparing the results of text-nontext classification, three different type of classifier are trained namely Naive Bayes, SVM with linear kernel and KNN. The extracted features from the previous step are used in a classifier, which classifies the text and nontext regions. Experiment result shows that KNN classifier outperforms the other two in F-Score. The nontext regions are then removed from the over all detected regions. The left regions in LR image are detected text regions.

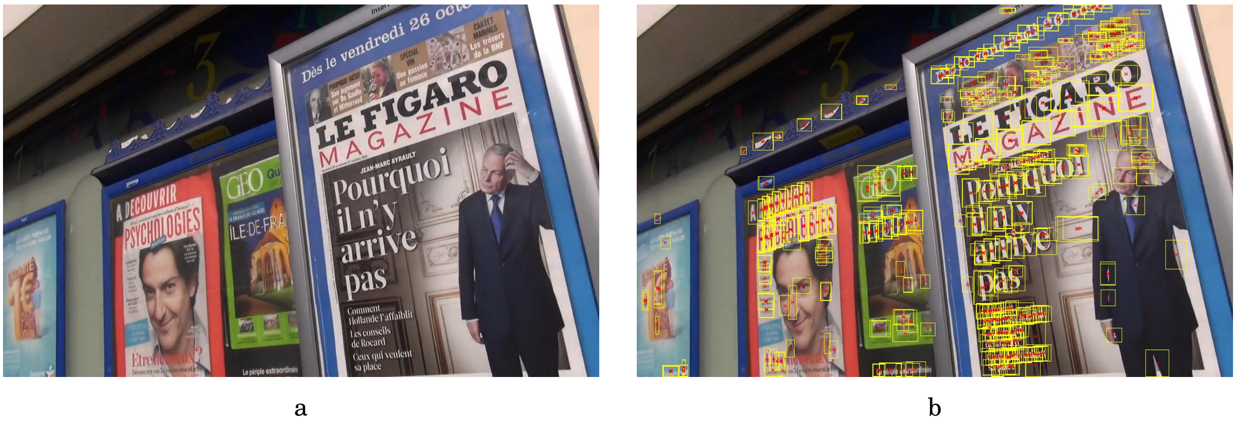
Chapter 4

Experiment Results

The code scripts are executed on Intel Xeon processor with 3.50 GHz and 32GB of RAM. We have used ICDAR 2015 video dataset for experimentation which contains single shot videos from single and moving camera.

4.1 Text Detection Results

The figure 4.1a represents the key frame of a video form ICDAR 2015 dataset while figure 4.1b represents overall text regions detected in the key frame. In figures, red star (*) is the centroid of the detected region.



a

b

Figure 4.1: Text Detection Example.In this figure, (a) shows a original frame obtained from a test video, and (b) shows overall detected text regions in the key frame.

Figure 4.2 compares the result of text detection for a certain part of the image, in between LR image and HR image. The images are zoomed so that a better representation can be presented. Figure 4.3 presents the comparison of text detection in LR image and

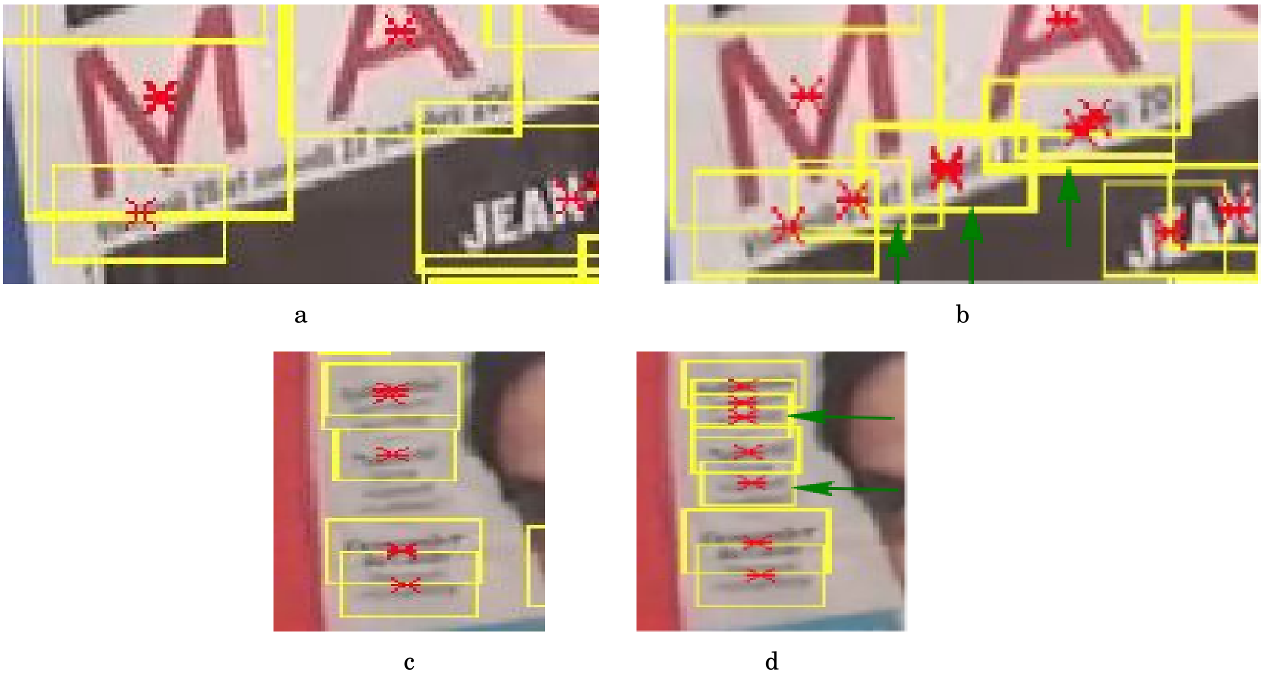


Figure 4.2: Comparison of text detection in LR image and HR image of key frame shown in figure 4.1a. In this figure, (a) shows a part of detected regions in LR Image, (b) shows detected regions in HR Image of the same part as in (a), (c) shows another part of detected regions in LR Image, (d) shows detected regions in HR Image of the same part as in (c).

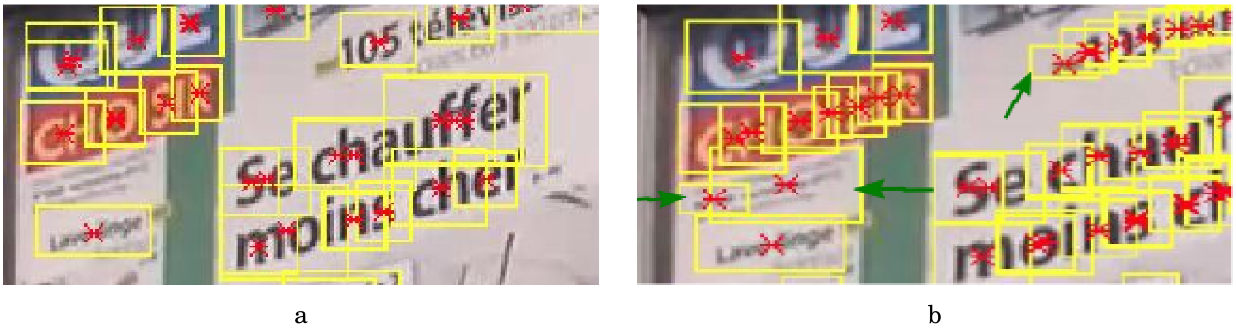
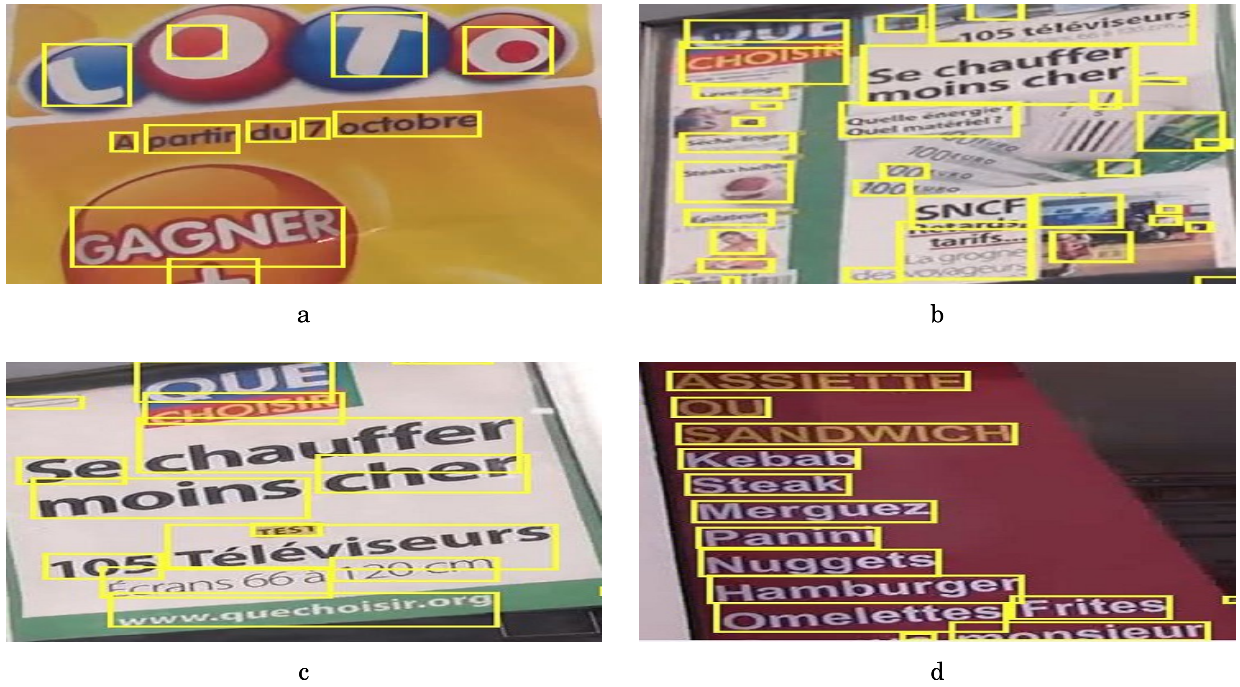


Figure 4.3: Comparison of text detection in LR image and HR image of another key frame. In this figure, (a) shows a part of detected regions in LR Image, (b) shows detected regions in HR Image of the same part as in (a).

HR image of another key frame in another video. Figure 4.4 shows result of text detection by proposed method on ICDAR 2015 dataset and figure 4.5 shows result of text detction on IITR video dataset.



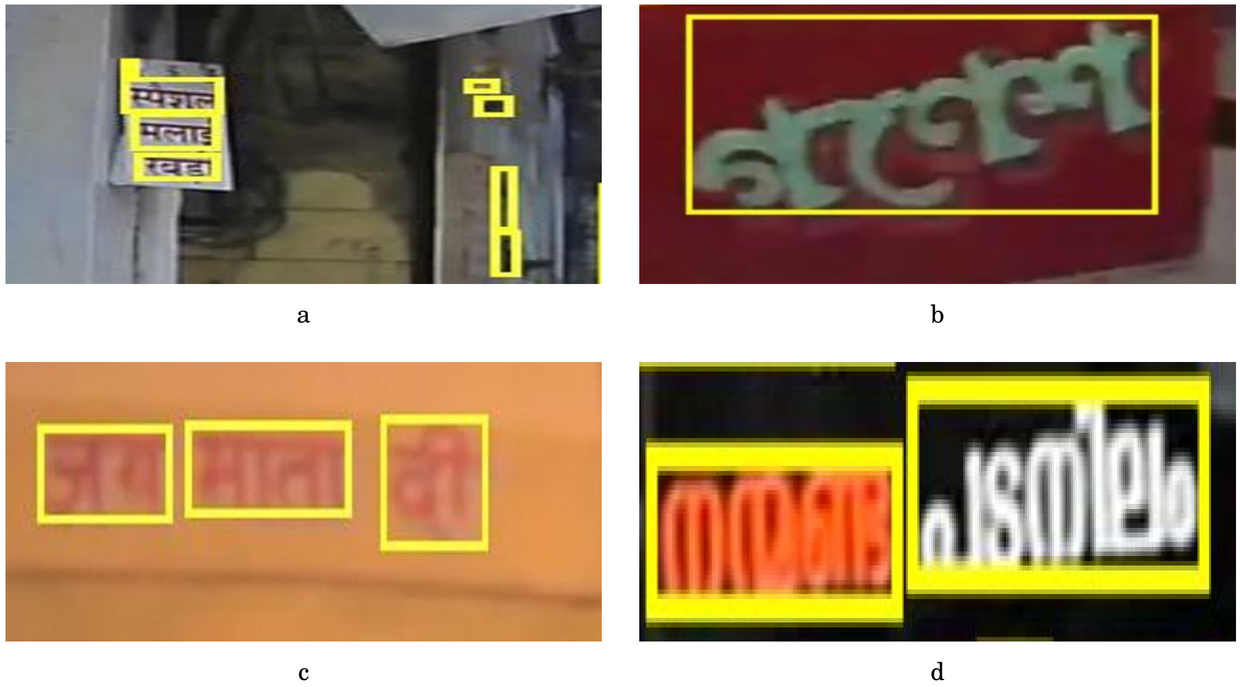
a

b

c

d

Figure 4.4: Text Detection Results on ICDAR dataset



a

b

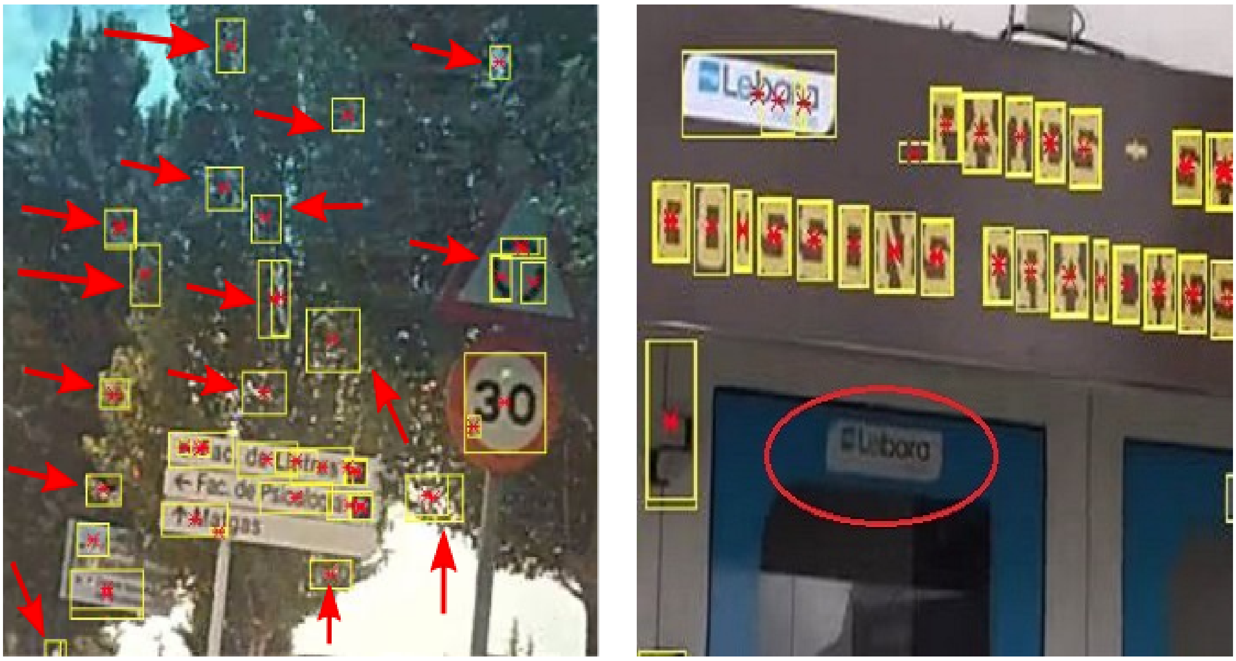
c

d

Figure 4.5: Text Detection Results on IITR dataset

4.2 Problems Associated with Text Detection

Figures in section 3.1 show that we have robust detection of text. As we are using MSERs detector to detect text in video frames, there is a chance that some text is missed in detection process of both HR image and LR image due to lighting effect. Also, MSERs may detect regions which do not contain text. And in our process, these nontext regions may increase as we are mapping detected regions from LR image and HR image. With the increase in accuracy of text detection, there is side effect also. Figure 4.6a and figure 4.6b shows the problem of detection of nontext regions and problem of missed out text respectively.In figure 4.6a, red arrows point to nontext regions and in figure 5(b), red ellipse shows missed out text.



a

b

Figure 4.6: Existing Problems in text Detection. In this figure, (a) represents detection of Non Text regions, and (b) represents missed Text region in a video frame.

4.3 Classification Results of Text and Non-Text using Feature Extraction

For classification of text and nontext regions, we took 1000 detected text regions randomly from different frames for classification and labeled them as text and nontext. We extracted HOG and LBP features from these regions. With 2x2 cell size for HOG feature and on the image size of 20 x 20, we are getting 2916 HOG feature and 59 LBP features from a single image. Next, we discuss how HOG and LBP features size are calculated,

4.3.1 Size of HOG features

The size of HOG feature vector depends upon few parameters such as cell size, block size, number of block in the image, and number of bins used.

- Size of images are 20 x 20 on which HOG features are calculated,
- With 50% overlapping, image is divided into 2 x 2 blocks where each block is a 2 dimensional matrix of cells.
- Each block consists of 2 x 2 cells and each cell consists of 2 x 2 pixels. Smaller cell size helps in collecting details at small scale. Also, smaller block size helps in removing effect of illumination changes.
- Number of blocks in the image will be 9 x 9. Also, gradient orientation is quantized into 9 bins.

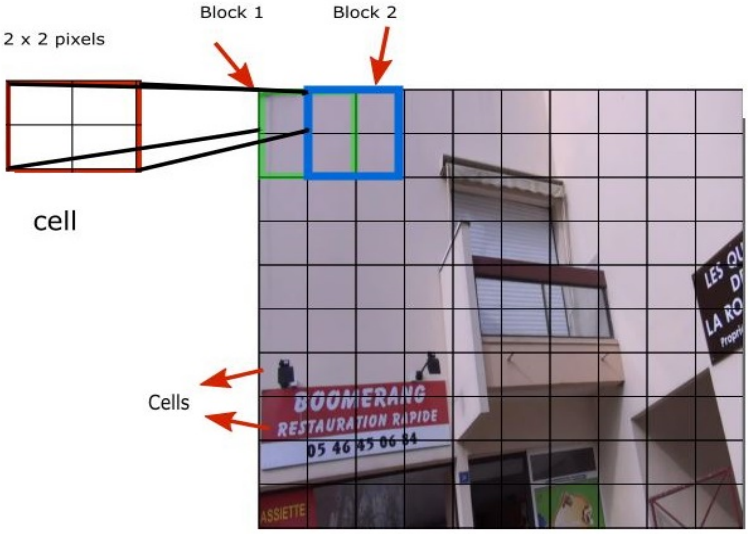


Figure 4.7: HOG features's size calculation in an image

Figure 4.8 shows how cell and block is represented in an image of size 20 x 20. Base image in figure 4.8 is a frame of a video taken from ICDAR 2015 dataset. As, number of HOG features is given by,

$$\begin{aligned} \textit{Number of HOG features} &= \textit{block size} * \textit{number of bins} \\ &\quad * \textit{Number of blocks in the image} \end{aligned}$$

Therefore, HOG features length with image size 20 x 20 will be,

$$\begin{aligned} \textit{Number of HOG features} &= 4 * (9 * 9) * 9 \\ &= 2916 \end{aligned}$$

4.3.2 Size of LBP features

The number of LBP features is the product of number of cells in a image and number of bins. In LBP extraction each pixel is surrounded by a circular symmetric pattern, from which neighbors are selected.

- Cell size ie taken equal to the image size. Therefore, number of cell is 1.
- Also, (8,1) neighborhood with 3 x 3 block size is considered.

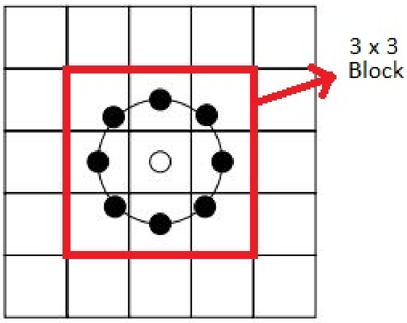


Figure 4.8: (8,1) neighborhood in LBP features calculation. Here 8 is number of neighbors around the center pixel while 1 is radius from center pixel.

Number of basic LBP features is given by,

*Number of LBP features = cell size **

*(number of neighbors * (number of neighbors-1) +3)*

Therefore, LBP features length with image size 20 x 20 will be,

$$\textit{Number of LBP features} = 1 * (8 * 7 + 3) = 59$$

Thus, total number of features calculated from a single bounding box is $2916 + 59 = 2975$. But these features are unnormalized. Therefore, we normalize these features using z-scores method. For this purpose, mean and standard deviation of each feature vector is calculated and then the feature is replaced with calculated new value. Let x' be the new feature vale and x is old feature value. X represents a particular feature vector. mean is represented by μ and σ represents standard deviation from the mean. Then the z-scores, also called as standard scores represented by x' are calculated as,

$$x' = \frac{x - \mu}{\sigma}$$

4.3.3 Analysis of Results of Classifiers

For classification of Text and Non Text regions, a total of 1000 MSERs regions are taken for labeling as Text and Non Text. We extracted HOG and LBP features from these regions and trained multiple classifiers. 70 % of data is used for training and 30 % is used for testing purpose. Figure 4.9 shows a bar chart representing the percentage of precision, recall, and F-score with LBP features only, HOG features only, and combination of both of these features applied to a linear SVM. On analyzing the bar chart, we can say that HOG feature perform well in precision, recall and F-score with respect to LBP feature. But when both features are combined, we get highest precision, recall, and F-score value i.e. 93%, 91%, and 92% respectively. Figure 4.10 shows a similar bar chart with features applied

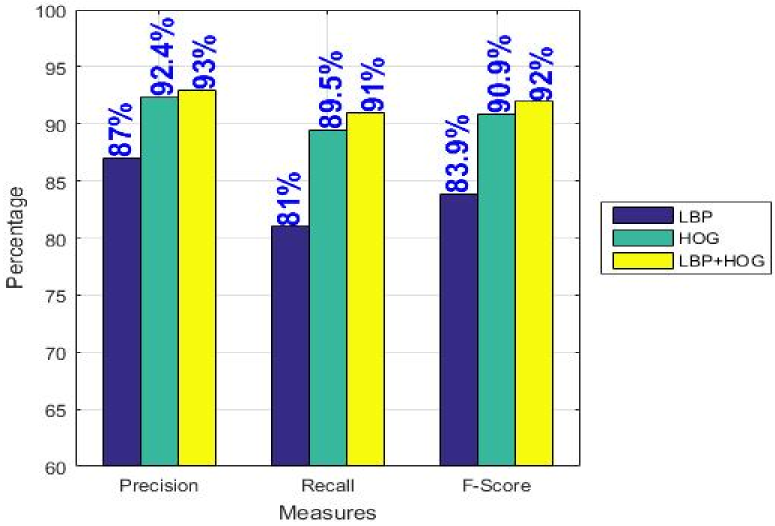


Figure 4.9: Bar chart showing precision, recall, and F-score of using different features with SVM classifier

to Naive Bayes classifier. On analyzing the bar chart, we can say that only LBP perform good in precision in comparison to only HOG and LBP+HOG (LBP and HOG combined), while in recall and F-score it performs poorer than boht only HOG and LBP+HOG. Also, only HOG and LBP+HOG give similar result in precision, recall and F-score with Naive Bayes classifier. Figure 4.11 also shows a similar bar chart with features applied to KNN

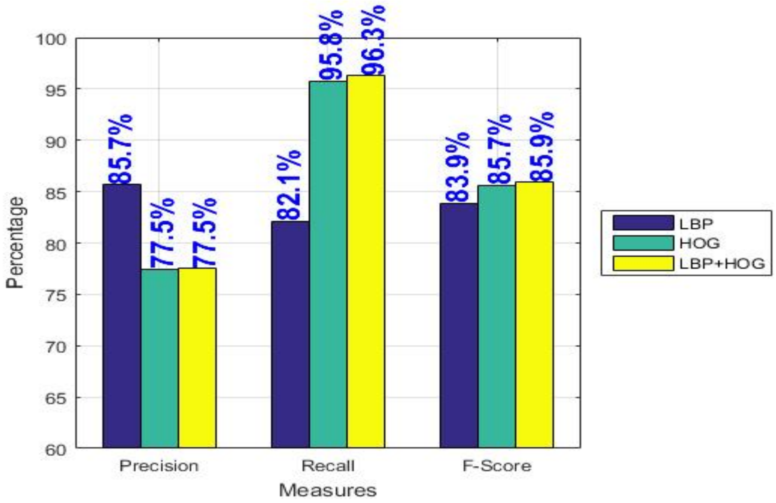


Figure 4.10: Bar chart showing precision, recall, and F-score of using different features with Naive Bayes classifier

classifier. On analyzing the bar chart, we can say that only LBP performance is not good with KNN. While LBP+HOG features always perform better with respect to only LBP and only HOG. .

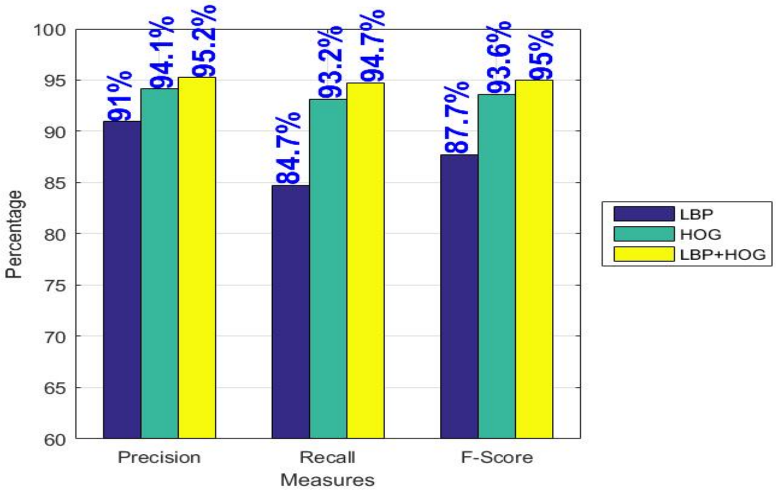


Figure 4.11: Bar chart showing precision, recall, and F-score of using different features with KNN classifier

4.3.3.1 Results with LBP+HOG features

As, results obtained by LBP and LBP features combined together are better, therfore we select LBP+HOG features for feature extraction from images. Following, we will discuss performnce of LBP+HOG features in detail. Table 4.1, Table 4.2, and Table 4.3 represents confusion matrix of Text and Non-Text class.

Table 4.1: Confusion Matrix for Naive Bayes Classifier

Class	Non-Text	Text
Non-Text	2	53
Text	7	183

In the case of Naive Bayes, recall is 96% but KNN outperforms others in case of precision and F-Score. There is stability in the measures in case of KNN. Table 4.4 shows the values of precision , recall, and F-Score for Naive Bayes, SVM, and KNN classifier for classification of text-nontext regions. Also calculation of precision, recall and F-score are done as following,

Table 4.2: Confusion Matrix for SVM Classifier

Class	Non-Text	Text
Non-Text	42	13
Text	17	173

Table 4.3: Confusion Matrix for KNN Classifier

Class	Non-Text	Text
Non-Text	44	11
Text	11	179

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

And F-score is harmonic mean of precision and recall and given by,

$$F - score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

where,

TP= total number of correctly detected text regions.

FP= total number of correctly detected nontext regions.

FN= total number of incorrectly detected text regions.

Table 4.4: Precision,recall and F-score values for classification of Text and Non-Text regions

Stats	Naive Bayes	SVM	KNN
Precision	77.5%	93%	95.2%
Recall	96.3%	91%	94.7%
F-score	85.9%	92%	95%

Figure 4.12 show analysis of the precision, recall, and F-score using bar charts. X-axis represents the different measures used for comparison like precision ,recall, and F-score. Classification technique Naive Bayes, SVM, and KNN are represented by bars. While Y-axis represents the percentage of recall, precision, and F-score respectively for classification techniques.

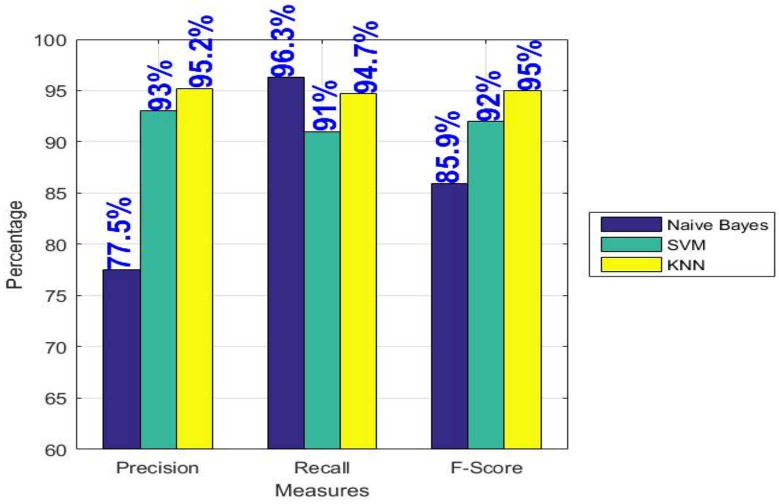


Figure 4.12: Bar Chart representation of precision, recall and F-score of different classifiers

4.4 Classification Results of Text and Non-Text using Convolutional Neural Network

In recent years deep learning is proved to be an efficient tool for various problems in machine learning and computer vision. Convolutional Neural Networks (CNN) is best suited for the problems related to images. CNNs are very similar to ordinary Neural Networks, as they are made up of neurons. And, these neurons have learnable weights and biases. In CNN, dot product is performed on each input recieved by neuron. But CNN still have a single score function which drives class score at the ned of the network.

In Regular Neural Networks, input (a single vector) is passed through a series of hidden layers. Each hidden layer is made up of a set of completely independent neurons and in single layer neurons do not share any connection but fully connected to all neurons in the previous layer. The last layer of model is output layer and it represents class scores. When Regular Neural Networks are used with full images, they do not scale well. Let an image of size 20 x 20 x 3 (20 width, 20 height, 3 color channels) is used with Rugalar Neural Nets, a single nuron in first hidden layer will have 20*20*3=1200 wights. The number of wights will be more complex when image of larger size is used. For example, an image of size 200 x 200 x 3 will lead to a neuron that have 120,000 wights. Furthermore, there are more number of neurons in a single hidden layer which add more wights which will be unmangeable. Clearly, this full connectivity is wasteful and the huge number of parameters would quickly lead to overfitting. Figure 4.13 shows a Regular Neural Network.

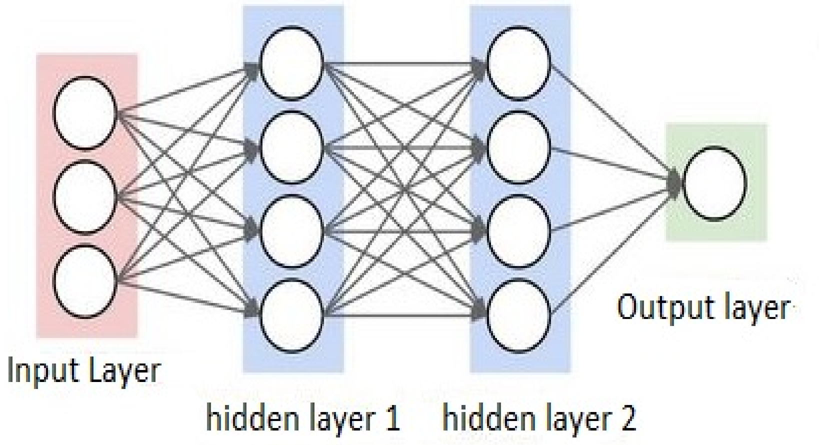


Figure 4.13: An Architecture of Regular Neural Network

On the other hand, Convolutional Neural Networks take advantage of architecture of images. Unlike a Regular Neural Network, the layers of a ConvNet have neurons arranged in 3 dimensions: width, height, depth. For example, the input images of size 20 x 20 x 3 will have a input volume of dimensios 20x20x3. Instead of fully connected manner of neurons, a neuron is connected to a small region of the previous layer. By the end of the ConvNet architecture, full image is reduced to a single vector of class score represented along the depth. Figure 4.14 shows the ConvNet architecture.

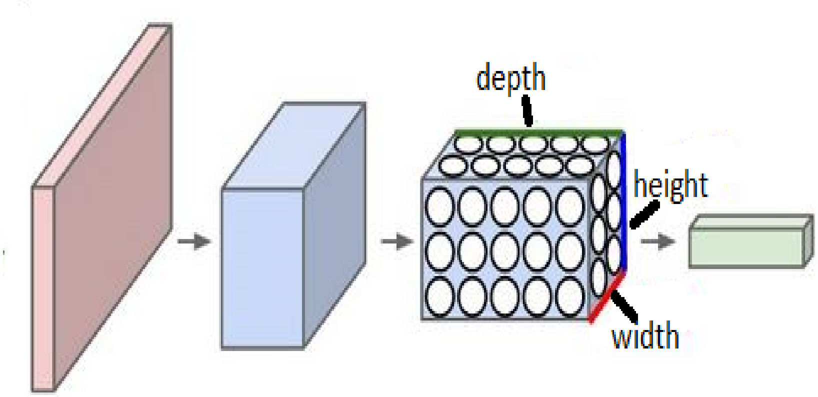


Figure 4.14: An Architecture of Convolutional Neural Network

A simple Convolutional Neural Network is made up of sequence of layer. And every layer uses a function to transform one volume of activations to another. Our CNN is based upon the VGGNet. We use three main types of layers to build ConvNet architectures: Convolutional Layer, Pooling Layer, and Fully-Connected Layer. These layers are stacked to form a full CNN.

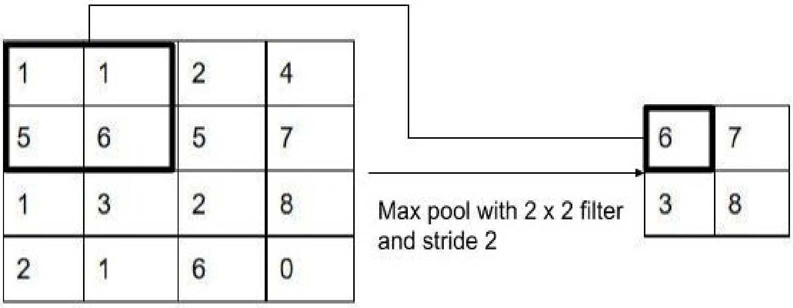
Input Layer: This layer holds the pixel values of images of size 20 x 20 x 1, with width 20, height 20, and grayscale image is used.

Convolutional layer: This layer computes the output of neurons using the local regions in the input. We are using 32 filters for the first layer and this results in the volume 20 x 20 x 32.

Activation layer: We are using ReLU as activation function. This layer perform ReLU elementwise which fires neurons only with positive weights. ReLU function is given by f(x)= max(0,x). This leaves the size of the volume unchanged ([20 x 20 x 32]).

Pooling layer: Pooling layer is used for subsampling and prevents from overfitting. Pooling function used is maxpooling. Maxpooling selects a maximum value from a defined neighbourhood. This layer will perform a subsampling only along the spatial dimensions, and results in volume [10 x 10 x 32]. Figure 4.15 shows a maxpooling process on an image.

Fully-connected layer: This layer will compute the class scores, resulting in volume of

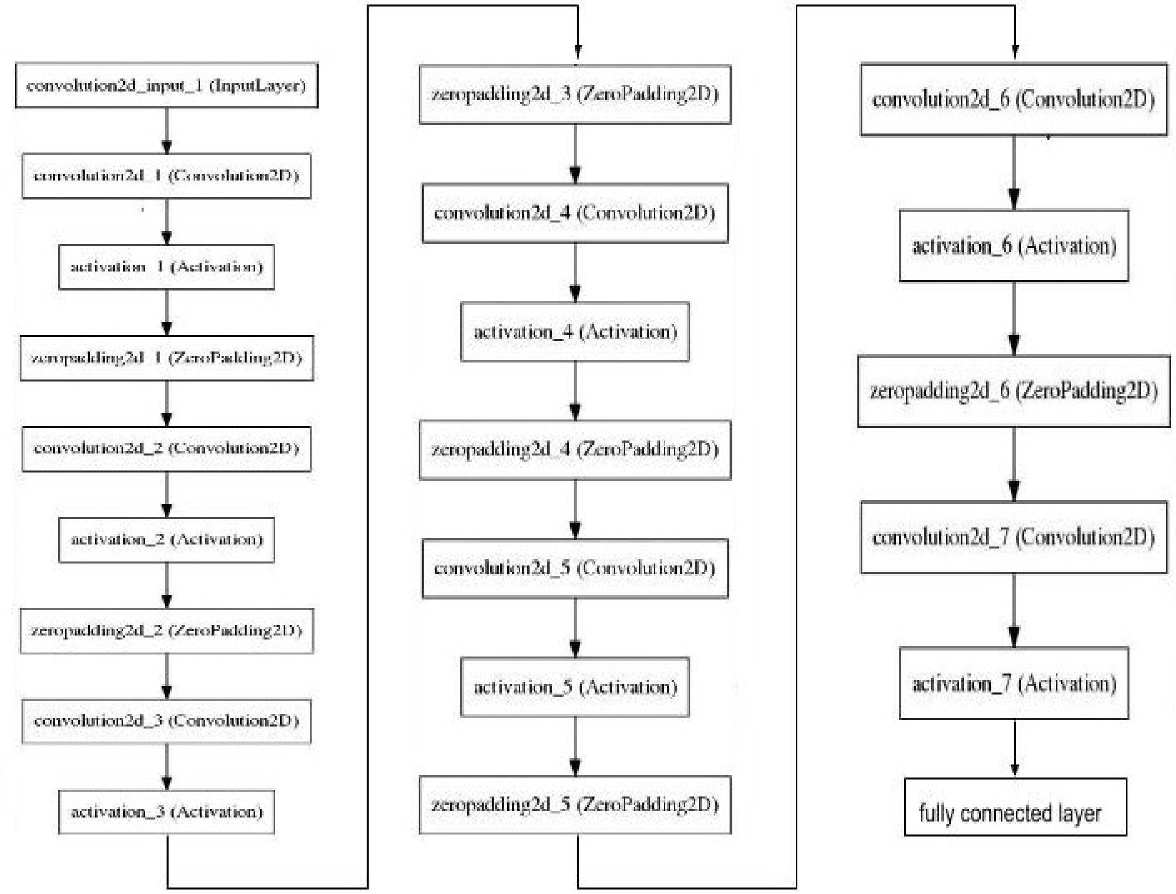


1	1	2	4
5	6	5	7
1	3	2	8
2	1	6	0

6	7
3	8

Figure 4.15: Maxpooling process on an image

size [1x1x2], where each of the 2 numbers correspond to a class score, representing text and non-text. In this layer each neuron is connected to all neorons of previous layer as in Regular Neural Networks.



convolution2d_input_1 (InputLayer)
convolution2d_1 (Convolution2D)
activation_1 (Activation)
zeropadding2d_1 (ZeroPadding2D)
convolution2d_2 (Convolution2D)
activation_2 (Activation)
zeropadding2d_2 (ZeroPadding2D)
convolution2d_3 (Convolution2D)
activation_3 (Activation)
zeropadding2d_3 (ZeroPadding2D)
convolution2d_4 (Convolution2D)
activation_4 (Activation)
zeropadding2d_4 (ZeroPadding2D)
convolution2d_5 (Convolution2D)
activation_5 (Activation)
zeropadding2d_5 (ZeroPadding2D)
convolution2d_6 (Convolution2D)
activation_6 (Activation)
zeropadding2d_6 (ZeroPadding2D)
convolution2d_7 (Convolution2D)
activation_7 (Activation)
fully connected layer

Figure 4.16: Proposed CNN model

Detected bounding boxes have different sizes. Firstly each box is resized to a 20 x 20 image. Then for the simplicity we convert them into a gray scale image. In our CNN model, there is one input layer of 20 x 20 images, three convolutional layer with 32 feature map and 3 x 3 kernel size, a maxpooling layer with filter size 2 x 2 and and again 4 convolutional layer with 64 feature map and finally a fully connected layer for computing class score for text and non text. Each hidden layer is equiped with ReLU for rectification and zero padding so that image size remains same for each hidden layer and all information is preserved through the convolutional process. Architecture of our CNN model is given in figure 4.16.

We have experimented with different convolutional layers in our model. And, the model consisting of 1 input layer, 3 convolutional layers, 1 pooling layer , 4 convolutional layers and a fully connected layer gives the best accuracy in all the models. In table 4.5, we have shown various configuration of ConvNet used for testing purpose.

Table 4.5: ConvNet Variations

ConvNet Configuration						
A	B	C	D	E	F	G
input layer (20 x 20 gray image)						
Conv2-32	Conv2-32	Conv2-32 Conv2-32	Conv2-32 Conv2-32	Conv2-32 Conv2-32 Conv2-32	Conv2-32 Conv2-32 Conv2-32	Conv2-32 Conv2-32 Conv2-32 Conv2-32
maxpooling (2x2 filter)						
Conv2-64	Conv2-64 Conv2-64	Conv2-64 Conv2-64	Conv2-64 Conv2-64 Conv-64	Conv2-64 Conv2-64 Conv-64	Conv2-64 Conv2-64 Conv-64 Conv2-64	Conv2-64 Conv2-64 Conv-64 Conv2-64
fully connected layer (FC 1024))						
fully connected layer (FC 512))						
fully connected layer (FC 2))						
soft-max						

These above configurations of ConvNet is used on the dataset of bounding boxes and separate text and non-text. For every configuration precision, recall, f-score and accuracy of model is calulated as shown in Table 4.6. From table we can conclude that model A gives best recall but model F gives highest precision, f-score and accuracy.

Table 4.6: Precision, Recall, F-score and Accuracy of above CNN models

CNN Model	Precision(%)	Recall(%)	F-score(%)	Accuracy(%)
A	91.54	97.35	94.35	90.98
B	92.42	96.82	94.57	91.39
C	93.75	95.23	94.48	91.39
D	93.68	94.17	93.93	90.57
E	92.82	95.76	94.27	90.98
F	93.84	96.82	95.31	92.62
G	93.65	93.65	93.65	90.16

In next chapter, we will draw conclusion of our experiments and discuss what can be done in future to increase the effectiveness of this method.

CHAPTER 5

Conclusion and Future Work

In this paper, we presented an approach for text detection in videos using MSERs and Super Resolution. The method maps the detected MSERs regions in high-resolution image and low-resolution key frame for better accuracy. But these detected regions, also contain nontext regions. Therefore, we extracted HOG and LBP features from text regions and trained a classifier to remove nontext MSERs regions. We compared the result of text-nontext separation task performed by different classifier namely Naive Bayes, SVM and KNN. SVM and KNN classifiers show good result for recall, precision, and F-score. Results show that CNN is better for classification purpose. Although the proposed method gives good results but still needs further improvement. In fact, due to the color instability some text is not detected. Also the text and nontext regions classification can be improved further. In future we will try to enhance the accuracy of text detection and text-nontext classifier. Also we will try to recognize detected text, even very smaller ones.

References

[1] M. A. Smith and T. Kanade, “Video skimming for quick browsing based on audio and image characterization,” *Citeseer*, 1995.

[2] A. K. Jain and B. Yu, “Automatic text location in images and video frames,” *Pattern recognition, Elsevier*, vol. 31, no. 12, pp. 2055–2076, 1998.

[3] K. Jung, “Neural network-based text location in color images,” *Pattern Recognition Letters, Elsevier*, vol. 22, no. 14, pp. 1503–1515, 2001.

[4] D. Chen, J.-M. Odobez, and H. Bourlard, “Text detection and recognition in images and video frames,” *Pattern recognition, Elsevier*, vol. 37, no. 3, pp. 595–608, 2004.

[5] W. Huang, Y. Qiao, and X. Tang, “Robust scene text detection with convolution neural network induced mser trees,” in *Computer Vision, ECCV, Springer*, pp. 497–511, 2014.

[6] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, “Robust text detection in natural scene images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE*, vol. 36, no. 5, pp. 970–983, 2014.

[7] J. Zhang and R. Kasturi, “Extraction of text objects in video documents: Recent progress,” in *The Eighth IAPR International Workshop on Document Analysis Systems, IEEE*, pp. 5–17, 2008.

[8] C. Jung, Q. Liu, and J. Kim, “A stroke filter and its application to text localization,” *Pattern Recognition Letters, Elsevier*, vol. 30, no. 2, pp. 114–122, 2009.

[9] P. Shivakumara, T. Q. Phan, and C. L. Tan, “A laplacian approach to multi-oriented text detection in video,” *IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE*, vol. 33, no. 2, pp. 412–419, 2011.

[10] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk, and B. Girod, “Robust text detection in natural images with edge-enhanced maximally stable extremal regions,” in *18th IEEE International Conference on Image Processing (ICIP), IEEE*, pp. 2609–2612, 2011.

[11] C. Shi, C. Wang, B. Xiao, Y. Zhang, and S. Gao, “Scene text detection using graph model built upon maximally stable extremal regions,” *Pattern recognition letters, Elsevier*, vol. 34, no. 2, pp. 107–116, 2013.

[12] H. Li, D. Doermann, and O. Kia, “Automatic text detection and tracking in digital video,” *IEEE Transactions on Image Processing, IEEE*, vol. 9, no. 1, pp. 147–156, 2000.

[13] R. Lienhart and A. Wernicke, “Localizing and segmenting text in images and videos,” *IEEE Transactions on Circuits and Systems for Video Technology, IEEE*, vol. 12, no. 4, pp. 256–268, 2002.

[14] Y. Li and H. Lu, “Scene text detection via stroke width,” in *21st International Conference on Pattern Recognition (ICPR), IEEE*, pp. 681–684, 2012.

[15] D. Karatzas, S. R. Mestre, J. Mas, F. Nourbakhsh, and P. P. Roy, “Icdar 2011 robust reading competition-challenge 1: Reading text in born-digital images (web and email),” in *International Conference on Document Analysis and Recognition (ICDAR), IEEE*, pp. 1485–1490, 2011.

[16] A. Papoulis, “A new algorithm in spectral analysis and band-limited extrapolation,” *IEEE Transactions on Circuits and Systems, IEEE*, vol. 22, no. 9, pp. 735–742, 1975.

[17] R. Gerchberg, “Super-resolution through error energy reduction,” *Journal of Modern Optics, Taylor & Francis*, vol. 21, no. 9, pp. 709–720, 1974.

[18] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide-baseline stereo from maximally stable extremal regions,” *Image and vision computing, Elsevier*, vol. 22, no. 10, pp. 761–767, 2004.

[19] W. Wolf, “Key frame selection by motion analysis,” in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP-96), IEEE*, vol. 2, pp. 1228–1231, 1996.

[20] F. Dirfaux, “Key frame selection to represent a video,” in *International Conference on Image Processing (ICIP), IEEE*, vol. 2, pp. 275–278, 2000.

[21] Y. Zhuang, Y. Rui, T. S. Huang, and S. Mehrotra, “Adaptive key frame extraction using unsupervised clustering,” in *International Conference on Image Processing (ICIP), IEEE*, vol. 1, pp. 866–870, 1998.

[22] S. C. Park, M. K. Park, and M. G. Kang, “Super-resolution image reconstruction: a technical overview,” *Signal Processing Magazine, IEEE*, vol. 20, no. 3, pp. 21–36, 2003.

[23] K. Jung, K. I. Kim, and A. K. Jain, “Text information extraction in images and video: a survey,” *Pattern recognition, Elsevier*, vol. 37, no. 5, pp. 977–997, 2004.

[24] S. Roy, P. P. Roy, P. Shivakumara, G. Louloudis, C. L. Tan, and U. Pal, “Hmm-based multi oriented text recognition in natural scene image,” in *2nd IAPR Asian Conference on Pattern Recognition (ACPR), IEEE*, pp. 288–292, 2013.

[25] S. Roy, P. Shivakumara, P. P. Roy, and C. L. Tan, “Wavelet-gradient-fusion for video text binarization,” in *21st International Conference on Pattern Recognition (ICPR), IEEE*, pp. 3300–3303, 2012.

[26] M. Irani and S. Peleg, “Improving resolution by image registration,” *CVGIP: Graphical models and image processing, Elsevier*, vol. 53, no. 3, pp. 231–239, 1991.

[27] W. T. Freeman, T. R. Jones, and E. C. Pasztor, “Example-based super-resolution,” *Computer Graphics and Applications, IEEE*, vol. 22, no. 2, pp. 56–65, 2002.

[28] L. Neumann and J. Matas, “Real-time scene text localization and recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE*, pp. 3538–3545, 2012.

[29] H. Li and D. Doermann, “Superresolution-based enhancement of text in digital video,” in *15th International Conference on Pattern Recognition, IEEE*, vol. 1, pp. 847–850, 2000.

[30] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, “Advances and challenges in super-resolution,” *International Journal of Imaging Systems and Technology, Wiley Online Library*, vol. 14, no. 2, pp. 47–57, 2004.