Automatic Text Identification, Recognition and Physical Address Generation from Unstructured Video

PROJECT REPORT

by

P THAISEER
(SC13B175)

DEPARTMENT OF EARTH AND SPACE SCIENCES
INDIAN INSTITUTE OF SPACE SCIENCE AND TECHNOLOGY
THIRUVANATHAPURAM
JULY 2016

Automatic Text Identification, Recognition and Physical Address Generation from Unstructured Video

PROJECT REPORT

by

P THAISEER
(SC13B175)

DEPARTMENT OF EARTH AND SPACE SCIENCES
INDIAN INSTITUTE OF SPACE SCIENCE AND TECHNOLOGY
THIRUVANATHAPURAM
JULY 2016

BONAFIDE CERTIFICATE

This is to certify that this Project Report entitled "**Automatic Text Identification, Recognition and Physical Address Generation from Unstructured Video**" submitted to **Indian Institute of Space Science and Technology, Thiruvanathapuram**, is a bonafide record of work done by **P THAISEER** under my supervision from **09-01-2017** to **28-04-2017**.

Dr. Gorthi R K S Subrahmanyam
Associate Professor

Dr. Anandmayee Tej
Head of Department
Department Of Earth and Space Sciences

Place
Date

Declaration by Author

This is to declare that this report has been written by me. No part of the report is plagiarized from other sources. All information included from other sources have been duly acknowledged. I aver that if any part of the report is found to be plagiarized I shall take full responsibility for it.

P THAISEER
SC13B175

Place
Date

ACKNOWLEDGMENTS

Words would not suffice to explain the support and advices I received from my guide **Dr. Gorthi R K S Subrahmanyam**, Department of Avionics, for giving me an opportunity to work under his guidance which is invaluable. His unflinching support, suggestions, directions have helped in smooth progress of this project. He has been a constant source of inspiration in all possible ways for successful completion of my project.I thank him for all the care and concern he has given to me.

I express my heartfelt thanks to **Dr. V K Dadhwal**, Director, Indian Institute of Space Science and Technology, for spending his valuable time during discussions and mentoring me through this project. Without his immense help, completion of this project will still be a mile far.

Abstract

Applications like content-based image indexing, real-time robotic navigation have generated research works in the problem of text reading from natural images and videos. In this project, we propose an efficient method for text identification, recognition and physical address generation from an unstructured video. The method we propose takes advantage of the similarity between image edges and text for the text identification part (along with ***Maximally Stable Extremal Regions***), and ***CNN*** for text recognition part. After finding the possible text regions, these are classified into text and non-text regions using *cnn* classifier. These text regions are given to another *cnn* classifier, which will recognize the text. Connecting these texts with already available latitude and longitude of text region generates the physical address of the location with recognized texts. Experimental results on simple image dataset was proven to be very effective in text identification and recognition with an average of greater than **99%** of texts being identified and recognized. On our available dataset, since the images obtained from video are much complex, almost **50%** of normal texts (texts with average size, illumination and clarity) were recognized with an average of one word per image.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Natural scene images and videos contains a lot of texts which conveys valuable information about that scene. These texts has important roles in the video and images as it is an important parameter used for information retrieval systems. Often, these texts required to be automatically recognized and processed for valuable information. Through this project, we strive towards an efficient method that aids automatic text identification, recognition and physical address generation from an unstructured video.

For this project, we have 16 frames per second video of a street (total length - 38 seconds), along with this, the latitude and longitude of the camera which had taken the video was also obtained. Through this project, we are trying to achieve an efficient method that will aids the automatic text identification, recognition and physical address generation from this dataset. Automatic text understanding and physical address generation systems can be employed in various applications that are useful in our daily life. One such application is an intelligent translation system that recognizes/understands text and translate it into a physical address based on the latitude and longitude along the video with the recognized texts. This can be very useful if we need to track a video on a map and adding informations about places in the video on to the map.

There is a spurt of activity for development of techniques that are useful in text identification and recognition from natural scene images due to the proliferation of digital cameras and the great variety of potential applications, as well. Many methods support a different viewpoint for identifying and recognizing texts from natural scene images. Researchers have focused their attention on development of techniques for understanding written texts in scene images / videos. In the reported works [1, 2, 3, 4] on text identification and recognition from natural scene images / videos, none of those works pertain to generate anything related to the location of the image

/ video taken and scope exists for exploring such possibilities.

Natural scene images and videos usually suffer from low resolution and low quality, perspective distortion and complex background. Scene text is hard to detect, extract and recognize since it can appear with any slant, tilt, in any lighting, upon any surface and may be partially occluded. In our case, it's more difficult because the video contains scenes of a street with shop names which are of non-uniform baselines, size, font and color etc..

There are wide variety of techniques for text detection from images and videos in literature. These methods can be classified in to two based on their approach towards text region identification. They are 1) sliding window based approaches and 2) connected component based ones. The reported works [5, 6] are based on sliding window based approach. In this method, text is detected from a given scene image by shifting a window on to all locations in multiple scales. Recent works [3, 4] on scene text detection tend to utilize the connected component based method. In these approach, character regions are first extracted from the scene image, where these character regions are set of pixels sharing similar text properties. The method for extracting and identifying texts from natural scene images is presented in [10]. However, they used extremal regions method for extracting text regions and adaboost classifier [11] for classification purpose, their algorithm makes use of double threshold and hysteresis tracking for better recall rates. This project work is closely related to their [10] work, we tried to modify their approach by using MSER's [7] for possible text region extraction and CNN[8, 9] for classification and recognition purposes.

Even though there is a spurt of activity in the field of text identification and recognition from natural scene images, none of the works pertain to generate any information out of the location of the image / video taken and scope exists for exploring such possibilities. Through this project, we are trying to explore the possibility of information retrieval from an unstructured video making use if the location details (latitude and longitude) of the place of which the video is taken from. Our approach mainly includes 6 steps, they are: **i)** getting different frames from the video **ii)** extracting the possible character regions from the frames **iii)** classifying the extracted regions into text and non-text regions using *CNN* [8] **iv)** recognizing characters from the text regions classified **v)** converting the characters into words based on their location and **vi)** finally connecting the location details of each frames with the words to generate the physical address.

The work flow of our approach is shown in the fig 1.1

The remainder of this report is organized as follows: Chapter 2 will discuss about our initial approach we tried for text region extraction, the approaches we finally

adopted for text region extraction and recognition is explained in Chapters 3 and 4, results and discussions on our work is described in Chapter 5 and Chapter 6 concludes this report.

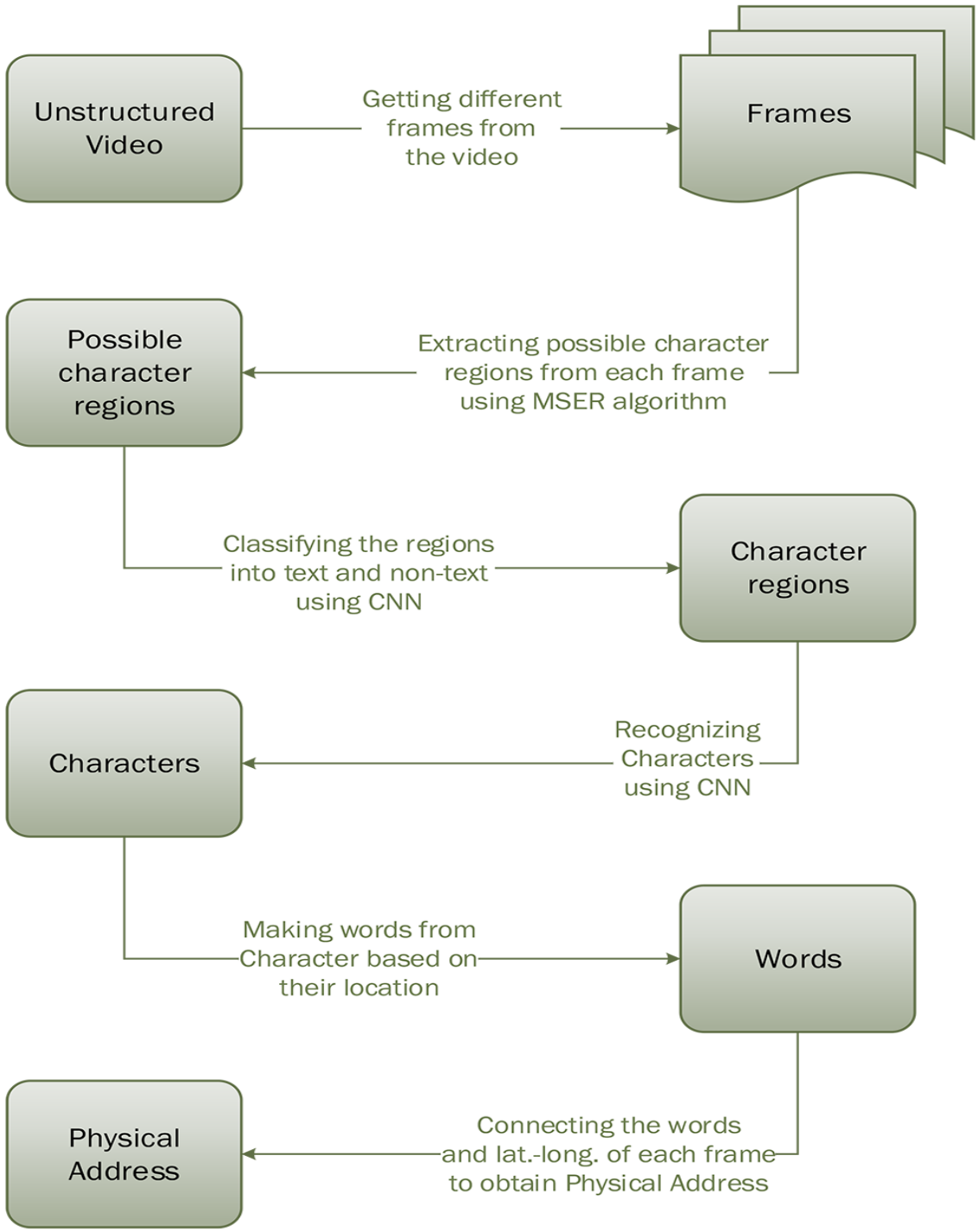


Figure 1.1: Work flow of our approach in this project

Chapter 2

Simple Image Processing Techniques

Natural scene images and videos usually suffer from low resolution and low quality, perspective distortion and complex background. Scene text is hard to extract since it can appear with any slant, tilt, in any lighting, upon any surface and may be partially occluded and in our case, it's more difficult because the video contains scenes of a street with shop names which are of non-uniform baselines, size, font and color etc. Even though the detection is a difficult task, we started our approach using simple image processing techniques like thresholding, edge detection, dilation, erosion etc.

2.1 Steps:

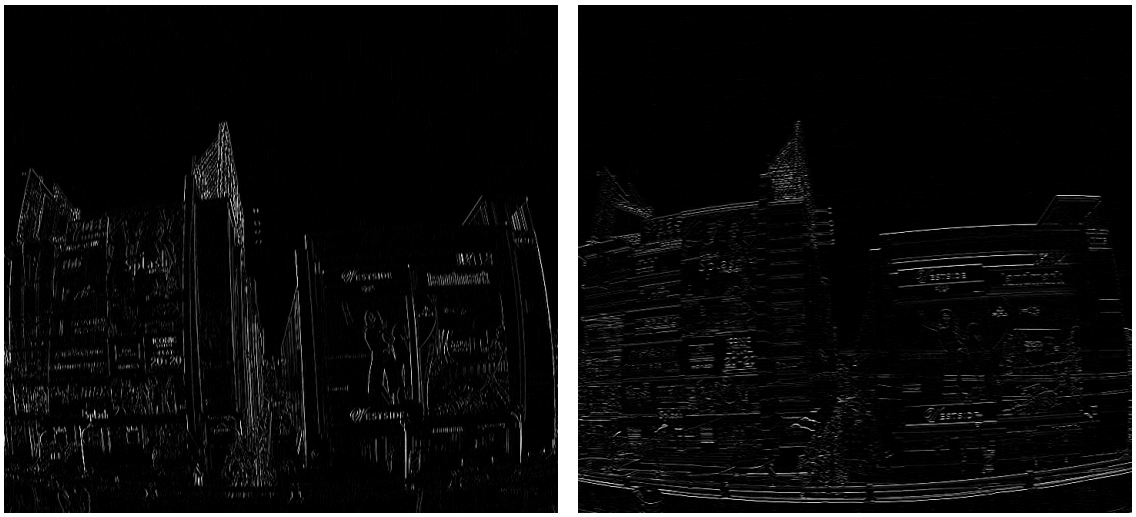
The approach we tried for possible text region extraction includes 5 steps. They are:

1. Detecting edges (both horizontal and vertical) separately.
2. Perform thresholding and dilation on both.
3. Take bitwise and operator of both the images.
4. Perform erosion followed by dilation.

These steps are pretty simple and straight forward approach for extracting any regions which are similar to image edges. Since the texts present in images will have many edges, we started with this approach. Detecting the edges and performing a threshold will get rid of all the background. Dilating these will enhance the horizontal edges in the horizontally edge detected image and on enhance the vertical edges in

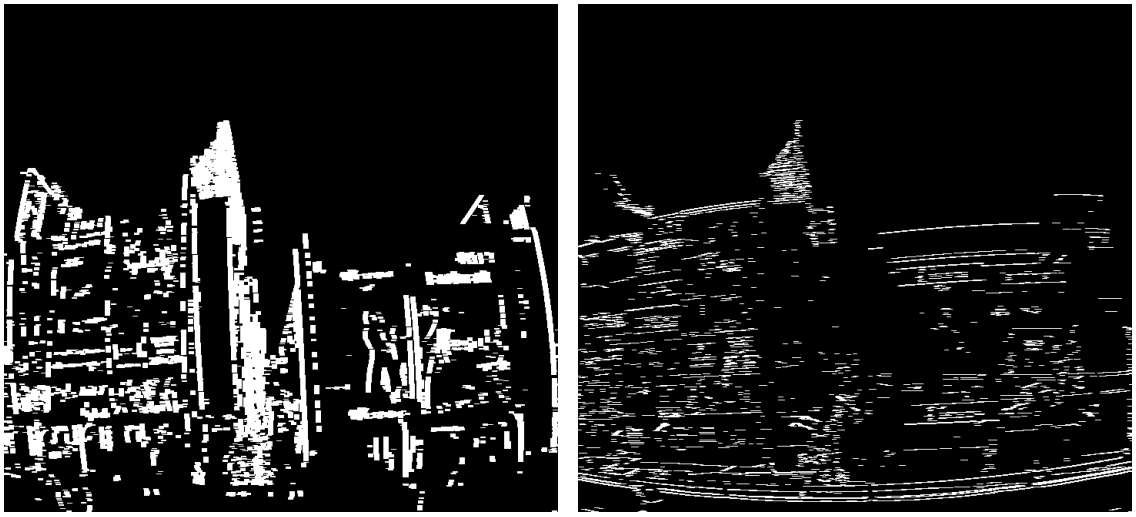
vertically edge detected images. Since in texts, it has both horizontal and vertical edges, on taking a bitwise and operation of these two enhanced edge image would result in possible text regions along with the corners of other edges. In order to remove the corners, we first do an erosion and then dilation to enhance the eroded regions. Since corners in the image which are remaining after bitwise and operation are small, they will get removed after this approach of erosion followed by dilation. Now we have possible text regions as our final result using simple image processing technique.
These initial approaches are shown in the figures 2.1 to 2.5.

Figure 2.1: Initial Frame



(a) Vertical (b) Horizontal

Figure 2.2: After applying edge detection



(a) Vertical (b) Horizontal

Figure 2.3: Thresholding and dilation applied on edge detected images

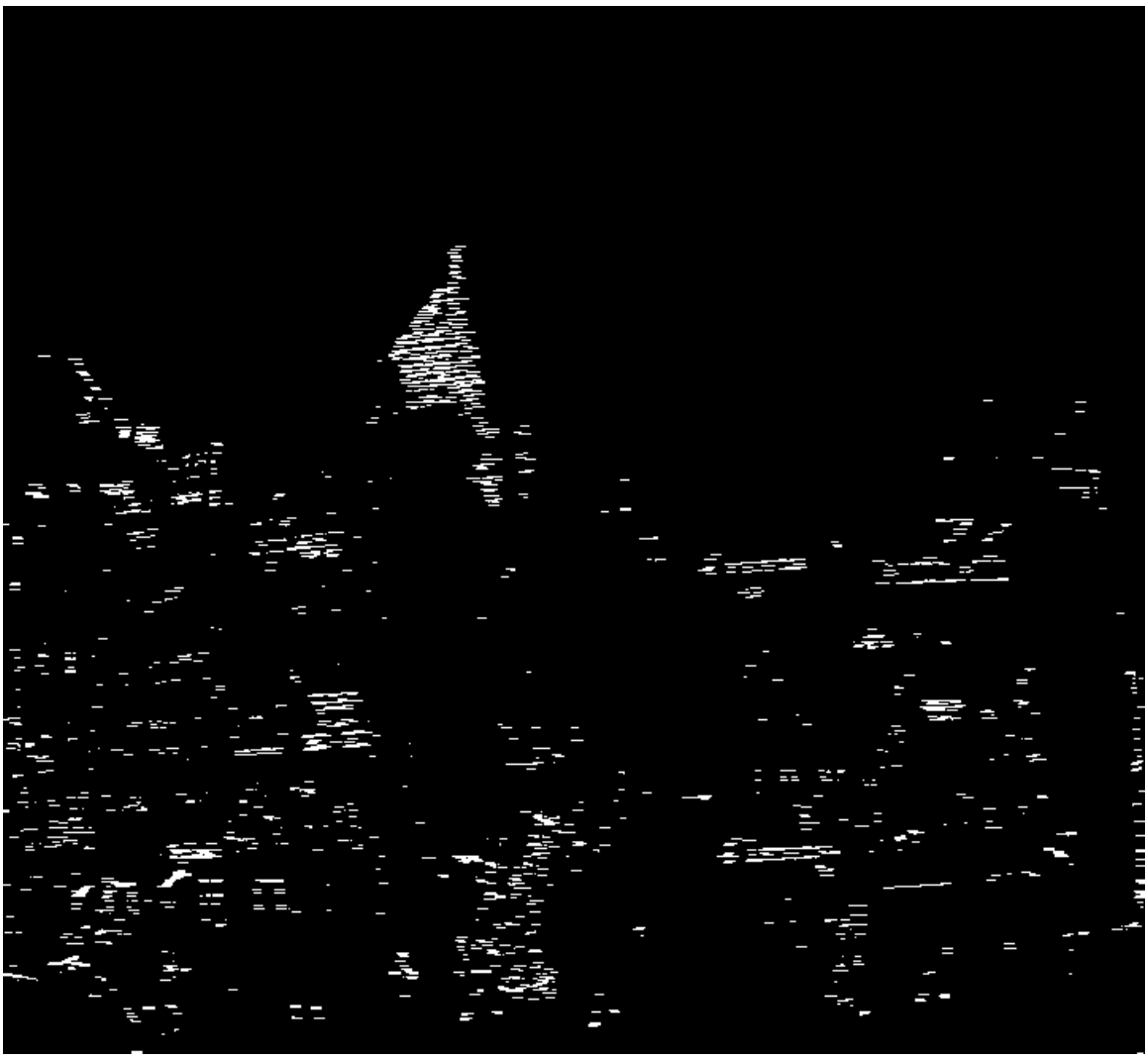
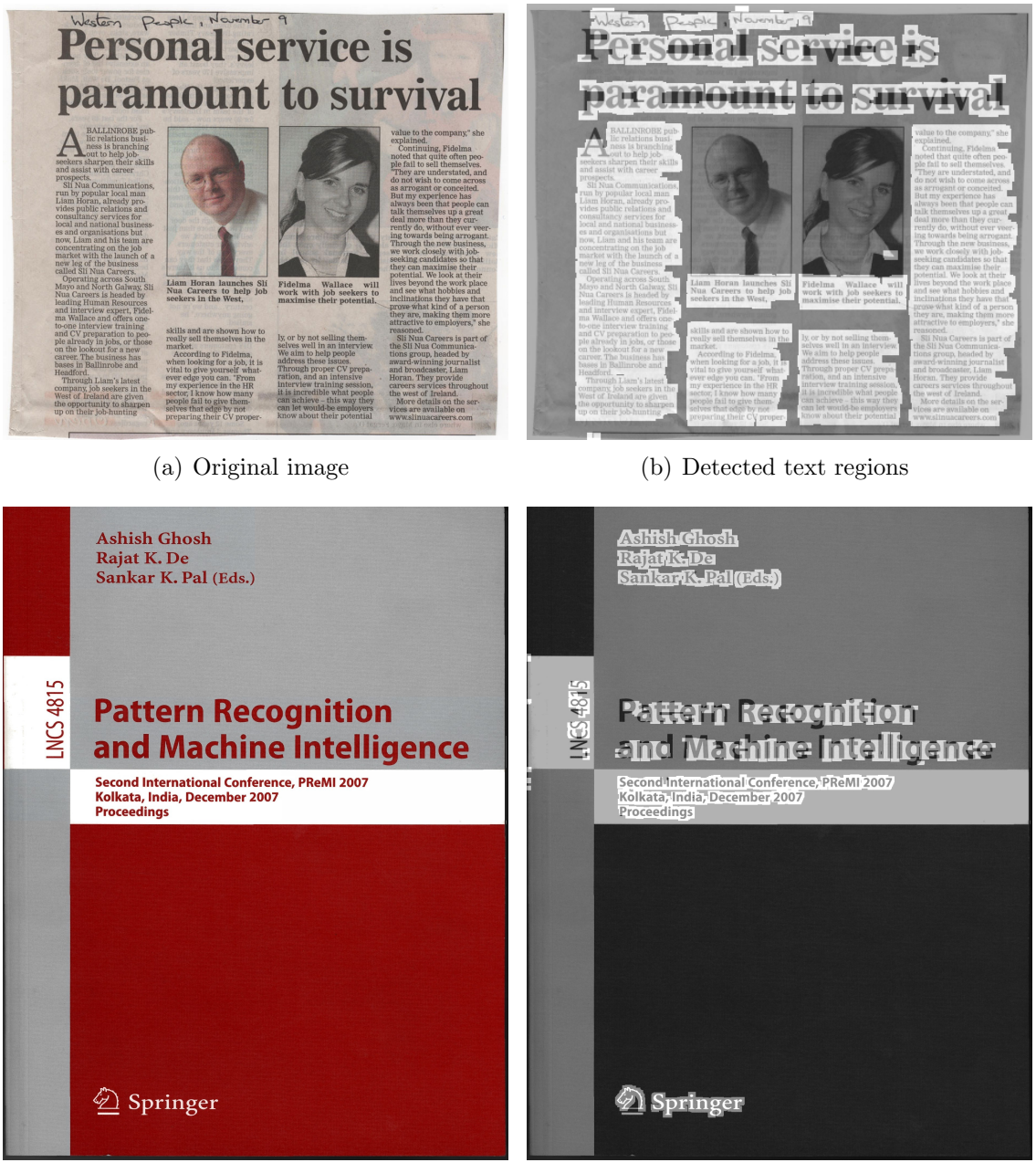


Figure 2.4: Bitwise and operated image of Vertical and Horizontal images

Figure 2.5: Performing erosion followed by dilation on image obtained in prev. stage

Even though the final image (fig. 2.5) produces a reasonably good result comparing that simple techniques on these scene images are not straight forward. It detects many text regions, but along with that, it produced many false positive regions also. Instead of using this scene image, if we use simple text images, the results were found to be greater than 90% success rate. Some of the examples are shown in fig 2.6.

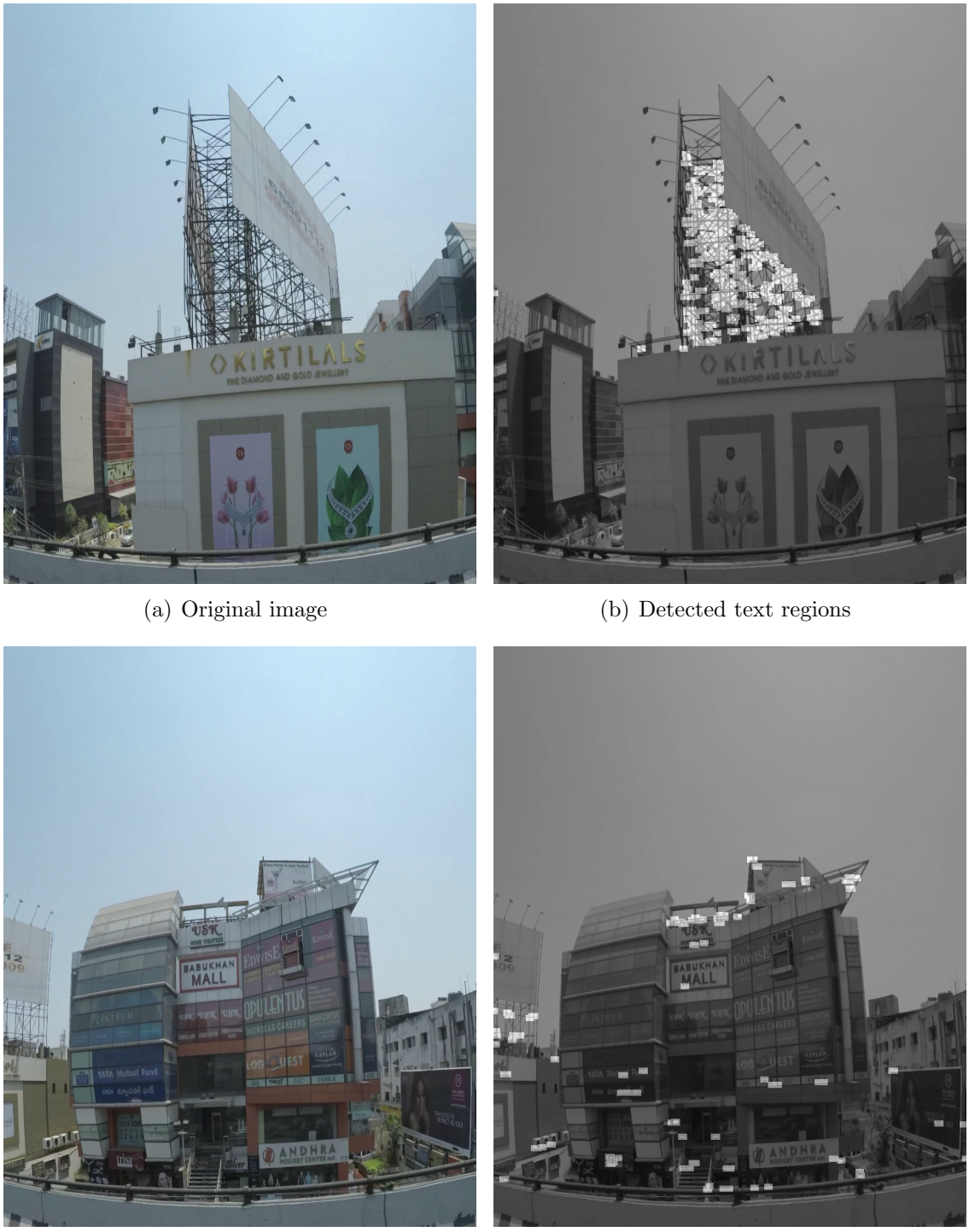


(a) Original image (b) Detected text regions

(c) Original image (d) Detected text region

Figure 2.6: Thresholding and dilation applied on edge detected images

Even though this simple image processing techniques worked for initial frame, it failed miserably for remaining frames, the results are shown in fig 2.7:



(a) Original image

(b) Detected text regions

(c) Original image

(d) Detected text region

Figure 2.7: Some frames for which this approach failed

Chapter 3

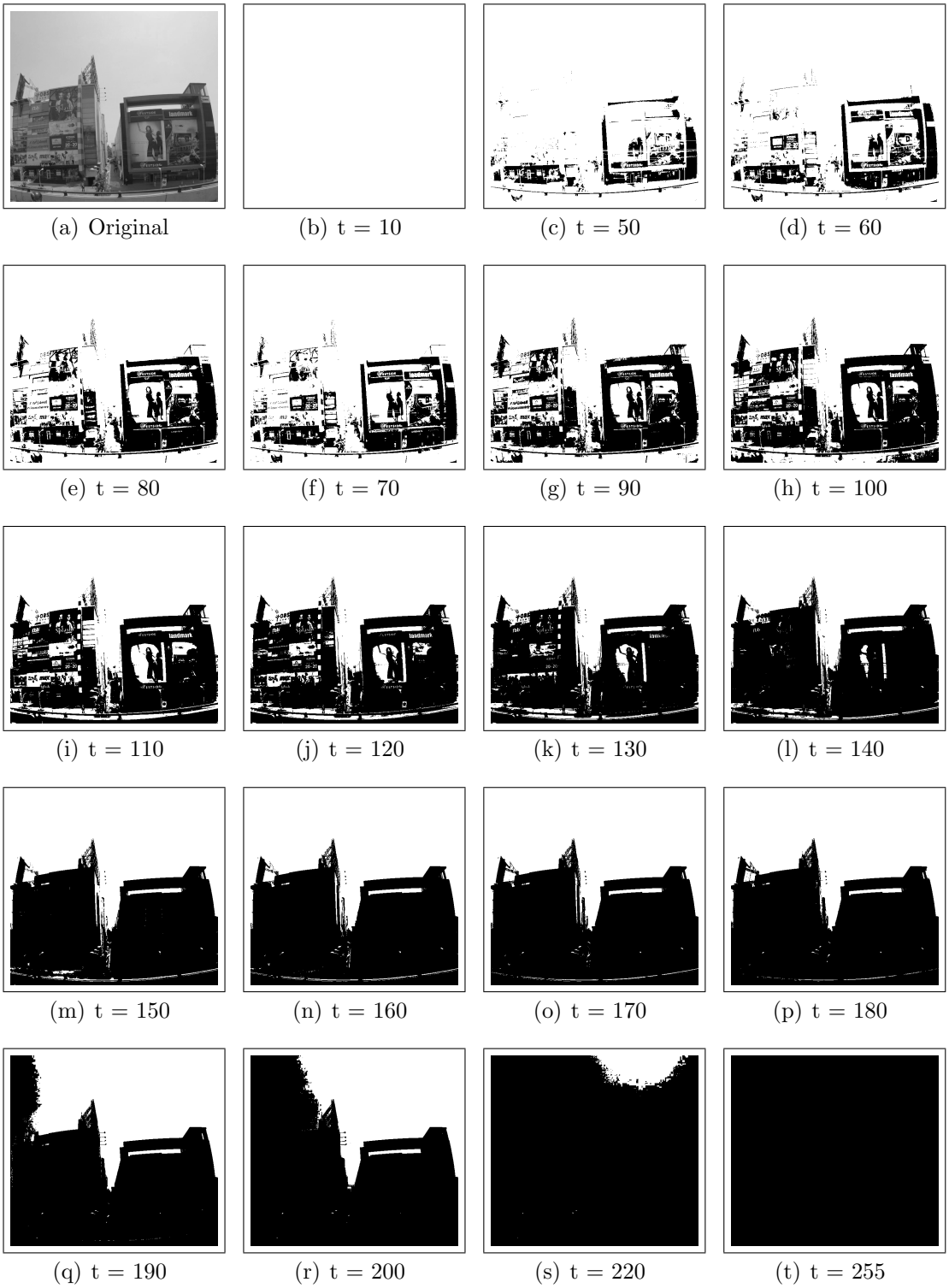
Maximally Stable Extremal Regions

Our images, which are natural scene images with with texts which are of non-uniform baselines, size, font and color etc. were not able to be detected using simple image processing techniques in all frames, so we came forward with a new technique called *maximally stable extremal regions* which is first put introduced by *Matas* and *Chum* in their work [7]. Their work was initially meant for wide baseline matching between pair of images taken at different view points, but in later stage, this got widely accepted as a feature extraction method in many problems.

3.1 What are MSERs? [7]

Maximally Stable Extremal Regions or *MSERs* are regions are defined solely by an extremal property of the intensity function in the region and on its outer boundary. Informally the concept of MSERs can be explained as follows. Imagine all possible thresholds of a gray-level image $\boldsymbol{I}$ (0 to 255). We refer to the pixels below a threshold t as black and those above or equal as white. If we are looking at all the thresholded images starting from *t=0* to *t=255*, we would see a white image first and subsequently black spots corresponding to local intensity minima will appear and grow, at some point, these will merge and finally we get a black image (shown in fig 3.1). The set of all connected components of all frames of the entire thresholded images is the set of all maximal regions.

MSER is based on the idea of taking regions which stay nearly the same through a wide range of thresholds. The word extremal refers to the property that all pixels inside the MSER have either higher (bright extremal regions) or lower (dark extremal regions) intensity than all the pixels on its outer boundary.



(a) Original (b) t = 10 (c) t = 50 (d) t = 60

(e) t = 80 (f) t = 70 (g) t = 90 (h) t = 100

(i) t = 110 (j) t = 120 (k) t = 130 (l) t = 140

(m) t = 150 (n) t = 160 (o) t = 170 (p) t = 180

(q) t = 190 (r) t = 200 (s) t = 220 (t) t = 255

Figure 3.1: Sweeping image thresholds

3.2 MSER Algorithm

- Pixels are sorted by intensity. (Order: n)
- After sorting, pixels are placed in the image (either in decreasing or increasing order).
- The list of connected components and their areas is tracked using the efficient union-find algorithm. (Order: $nlog(log(n))$)
- The process produces a data structure storing the area of each connected component as a function of intensity.
- Finally, intensity levels that are local minima of the rate of change of the area function are selected as thresholds producing maximally stable extremal regions.

3.3 Observations on simple images

For the purpose of testing the validity of MSER algorithm for text region extraction, we used natural scene images from ICDAR 2015 dataset [12]. On those images, the algorithm worked very well and found out almost all text regions present in those images along with some false positive regions which we need to eliminate. These are shown in fig 3.2 to 3.4.

Figure 3.2: MSERs initially identified on ICDAR 2015 images [12]

Figure 3.3: MSERs initially identified on ICDAR 2015 images [12]

Figure 3.4: MSERs initially identified on ICDAR 2015 images [12]

3.4 Observations on street video frames

On applying same algorithm on scene images from street video, prominent text regions are extracted and some with less luminosity and clarity are not detected in MSERs.

Fig 3.5 to fig. 3.7 shows this applied on some of the image frames from video.

Figure 3.5: MSERs initially identified on video frame 1

3.5 Filtering of MSERs

Before going for text and non text classification, we can eliminate many regions from the detected MSERs using simple techniques as follows:

- There are overriding regions which represent same regions, this can be eliminated using the position of each regions.
- There are regions in which the width by height ratio is very high, these can also be removed by keeping a threshold on width by height ratio.
- There single regions at random locations, text regions will occur with a uniform distance between the characters and at-least 2 regions together (we are assuming that there will not be any word in the image with only one character). These regions can also be removed using a comparison based on location of different regions.

Above steps applied on ICDAR dataset image is shown in fig. 3.8 and on street video image frame is shown in fig. 3.9. After removing all these regions, we are left

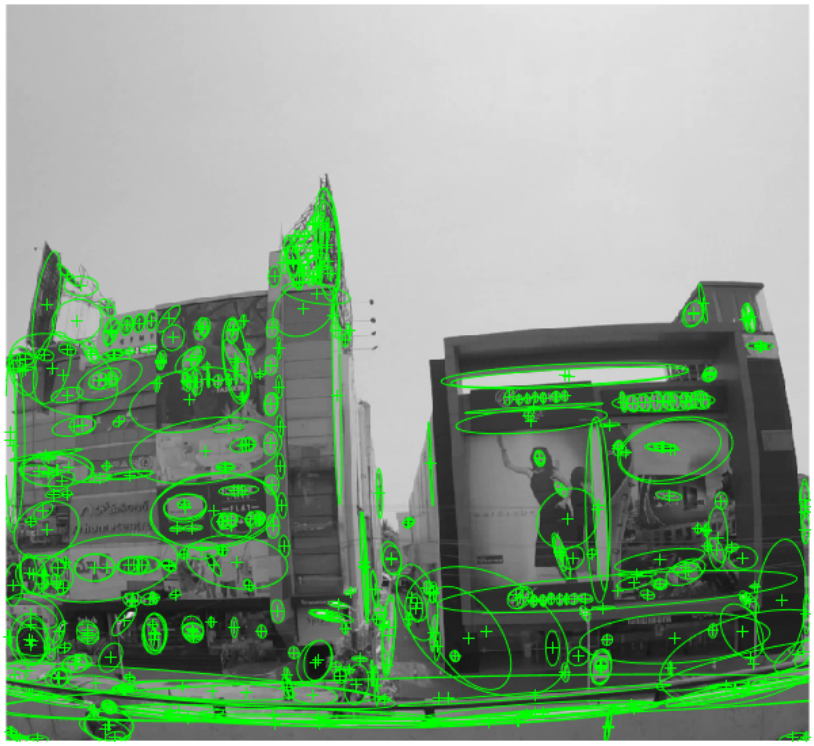


Figure 3.6: MSERs initially identified on video frame 2

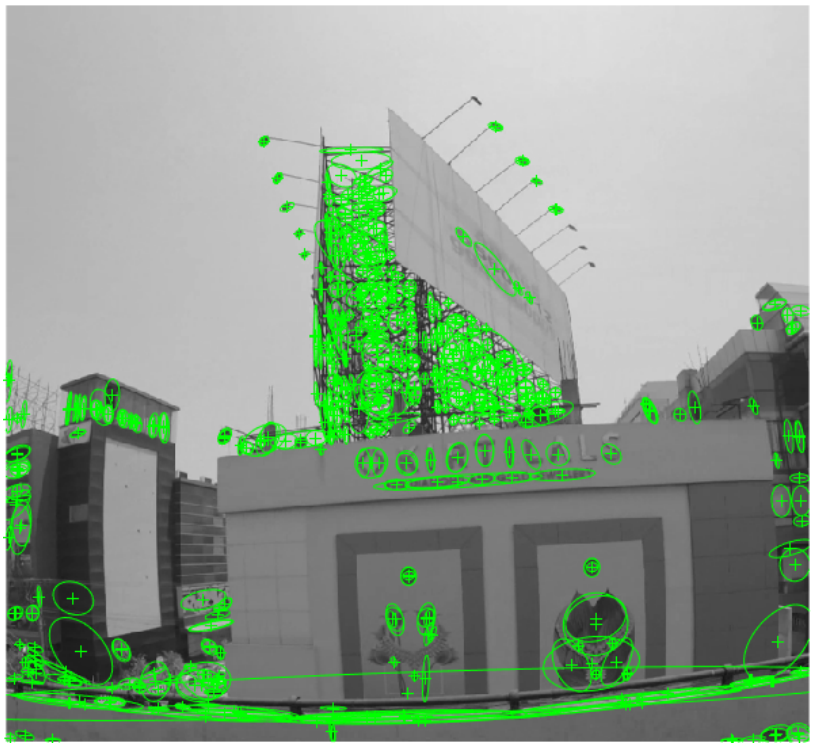


Figure 3.7: MSERs initially identified on video frame 3

with regions which includes some text and some non-text regions. These are classified using CNN explained in Chapter 4

Figure 3.8: Filtered MSERs (image from ICDAR dataset [12])

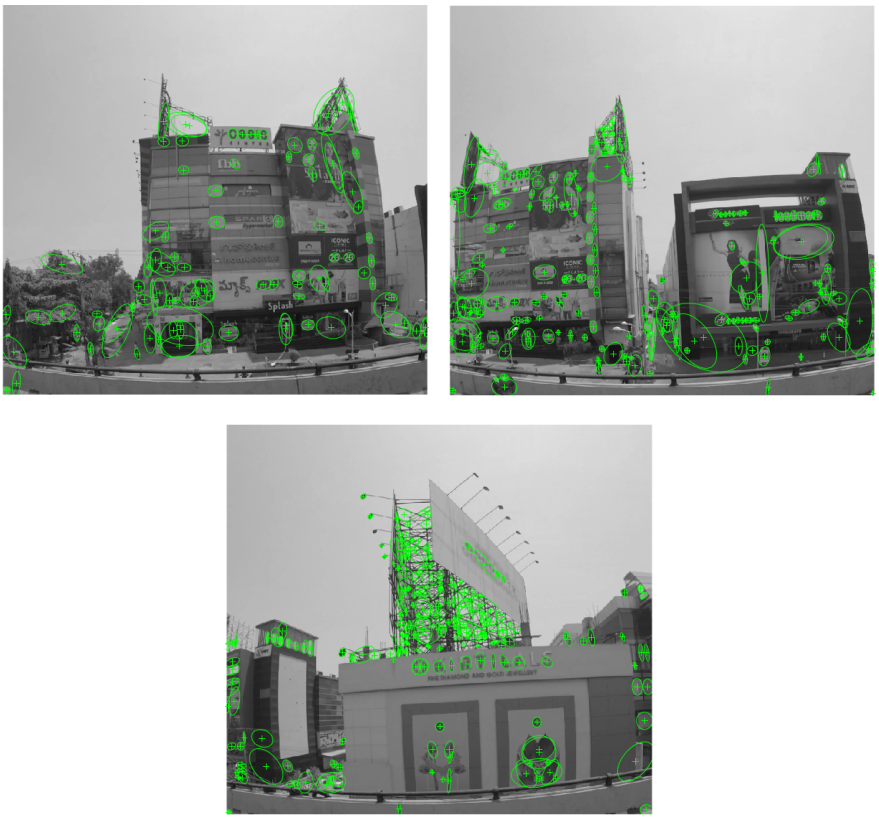


Figure 3.9: Filtered MSERs (Street video frames)

Chapter 4

Text Identification and Recognition

Once we obtain the filtered MSERs, we can apply classification techniques to classify the regions into text and non-text. Here we used **convolutional neural network** for classification purpose. After classification, the reminder regions had some false positive text regions (fig. 4.1 and fig. 4.3), these can be removed using further filtering. After final filtering, we have regions which are 100% texts, these can be recognized using another **CNN** classifier which is trained using character datasets created from texts extracted from ICDAR train dataset [12] modified according to our need.

4.1 Convolutional Neural Network (CNN) [8, 9]

Convolutional Neural Networks are very similar to ordinary feed-forward artificial neural networks, they are made up of neurons that have learnable weights and biases. The response of an individual neuron to stimuli within its receptive field can be approximated mathematically by a convolution operation, means- each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. In this project, we used two configuration of neural networks, one for classification of text and non-text regions and another one for recognizing the final texts identified.

4.1.1 Configuration of CNN for text and non-text classification [13]

For text and non-text classification, we used a convolutional neural network of following configuration:

- Image input layer of size 24x24.

- Convolutional layer with 20 filters, each with a height and width of 5.
- ReluLayer for thresholding
- Max pooling layer with non-overlapping pooling regions, which down-samples by a factor of 2.
- Fully connected layer with an output size of 2 and input size of 2000(which is the output size of preceding max pooling layer - 2000).
- Soft-max layer.
- Training options were: initial learning rate: 0.001, number of epochs: 20 and batch size as 50.

4.1.2 Configuration of CNN for text recognition [13]

For text recognition, we used a convolutional neural network of following configuration:

- Image input layer of size 24x24.
- Convolutional layer with 5 filters, each with a height and width of 11.
- ReluLayer for thresholding
- Max pooling layer with non-overlapping pooling regions, which down-samples by a factor of 2.
- Fully connected layer with an output size of 52.
- Soft-max layer.

4.2 Observations

Upon applying classification using *cnn* on the filtered MSERs, we got left with very few regions which are mostly text (fig. 4.1 and fig. 4.3). We can see still few false positives in these images, which can be further filtered using simple techniques. After classification, the positive text regions come clustered in same base-line for a word and the false positive comes randomly without any clustering (assuming that there is no single character text to be recognized, all are multi-character words). This property of text regions can be utilized for final filtering. The final filtered regions are shown in fig. 4.2 and fig. 4.4

(a) Before classification (b) After classification

(c) Before classification (d) After classification

(e) Before classification (f) After classification

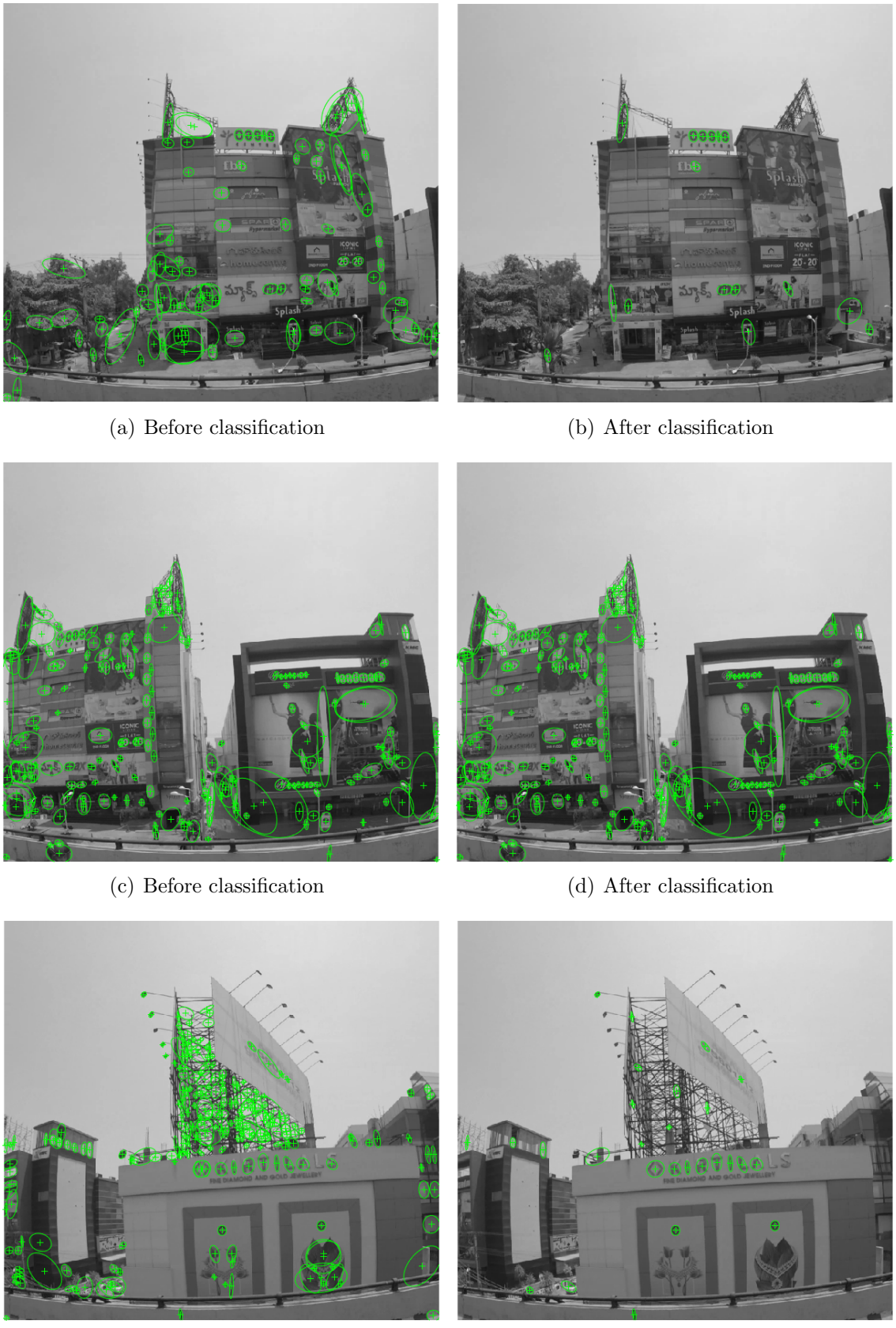
Figure 4.1: Text and non-text classification results using cnn on ICDAR images [12]

(a) Before filtering (b) After filtering

(c) Before filtering (d) After filtering

(e) Before filtering (f) After filtering

Figure 4.2: Final filtering results on ICDAR images [12]



(a) Before classification

(b) After classification

(c) Before classification

(d) After classification

(e) Before classification

(f) After classification

Figure 4.3: Text and non-text classification results using cnn on street video frames

(a) Before filtering (b) After filtering

(c) Before filtering (d) After filtering

(e) Before filtering (f) After filtering

Figure 4.4: Final filtering results on street video frames

The final filtered regions has only text in it and these can be given to our second cnn model which does the recognition part. After recognizing the text, characters are combined together based on their position to form words. The recognition part is shown in fig. 4.5.

4.3 Physical address generation for video frames

From the final identified text on street video frames, the physical address generation is done by connecting the frame number of text on which it appeared and the latitude and longitude of the frame which we already have.

Figure 4.5: Recognized text on ICDAR images (left) and street video frames (right)

Chapter 5

Results, Comparison and Conclusions

Initially we tried with some basic image processing techniques to identify the text regions. This technique worked reasonably well compared with the complexity of texts present in the video frames for some frames, but it failed completely for most of the frames. So we had to look for another technique for text region identification and we came up with MSER algorithm combined with cnn classification and some filtering. In order to check the validity of this algorithm, we took a standard images from ICDAR 2015 dataset. We applied the same for our video frames also, but in case of video frames, the quantification of results are not that straight forward. Since the video frame images are not completely luminous and in some cases some texts are too small for the algorithm to work, so in order to quantify, first we should define some criteria which tells us which all texts from the frame should be identified. The criteria which we defined is we will include all characters in the image which are atleast two character length and with reasonable clarity and luminosity. After doing this analysis, we found the following results:

Dataset	Identified texts	Recognized from identified regions
ICDAR images	> 99.5%	>99%
Street video frames	40% - 50%	>99%

Table 5.1: Results for ICDAR images and street video frames

From the table, it is clear that, the algorithm works very well for normal scene images which has very good clarity or focused texts in it. And the same worked to about 40% to 50% accuracy on street video frames. Low accuracy in the case of street video frames are due to reasons like low quality of image, unfocused text

regions, complexity of scene etc. In recognition part, we got almost 100% accuracy, but in few cases, *i* and *l* where misclassified mainly because MSER failed to capture the period on *i*.

5.1 Conclusions and Recommendations

Even though the result worked very well for ICDAR standard dataset, it went just average for street video frames which are not well focused images. In order to improve the same algorithm on these dataset as well, we can make use of **extremal regions** instead of **maximally stable extremal regions**. If we use **ERs** instead of **MSERs**, number of regions detected in first stage will be very high but we can eliminate most of them using filtering and finally classification. But working with large number of regions will be computationally difficult and time consuming task.

Appendices

Appendix A

Frame Extraction from video: Python Code

```
% File Name: getFrames.py

import cv2
vidcap = cv2.VideoCapture('../Files/video.avi')
count = 0
success = True
fps = vidcap.get(cv2.CAP_PROP_FPS)

while success:
    frameId = int(round(vidcap.get(1)))      # Getting current frames ID
    success, image = vidcap.read()
    if frameId == 0:                         # Saving initial frame
        print('Read a new frame: ' + str(frameId))
        cv2.imwrite("../Files/frames/%d.jpg" % count, image)
        count += 1
    elif frameId % fps == 0: # Making sure that only one image is saved
        per second
        print('Read a new frame: ' + str(frameId))
        cv2.imwrite("../Files/frames/%d.jpg" % count, image)
        count += 1

vidcap.release()
```

Appendix B

Region extraction from frame: Sample MatLab Code

```
image = imread('../Files/frames/frames/0.jpg');
image = rgb2gray(image);
if ((size(image,1) >=1000) && (size(image,1) <2000))
    image = imresize(image, 0.5);
elseif ((size(image,1) >=2000) && (size(image,1) <3000))
    image = imresize(image, 0.35);
elseif ((size(image,1) >=3000))
    image = imresize(image, 0.25);
end
[regions, ~] = detectMSERFeatures(image, 'RegionAreaRange',[20 5000] );
figure; imshow(image); hold on;
plot(regions, 'showEllipses', true);
title('Detected MSER Regions');
%
RegionsToRemove = RemoveOverlappingReg(regions, 4);
regions(RegionsToRemove==1) = [];

figure; imshow(image); hold on;
plot(regions, 'showEllipses', true);
title('After removing overlapping regions and regions based on aspect
    ratio');

regionscl = regions;
%
```

```
Size = [24,24];
load('convnet');
Label = GetLabel(image,regions,Size,convnet);

regions(Label ~= 2) = [];

figure; imshow(image); hold on;
plot(regions, 'showEllipses', true);
title('Regions after classification');

%
for n = 1:4
    RegionsToRemove2 = RemoveNonText(regions);
    regions(RegionsToRemove2==1) = [];
end
%
figure; imshow(image); hold on;
plot(regions, 'showEllipses', true);
title('FinalText Regions');
```

References

[1] Shivakumara,P. , Sreedhar,R.P. , Trung Quy Phan , Shijian Lu , Chew Lim Tan, *"Multioriented Video Scene Text Detection Through Bayesian Classification and Boundary Growing,"* IEEE Transactions on Circuits and Systems for Video Technology, Vol. 22 , No.8 pp. 1227-1235, 2012.

[2] Xiaoqing Liu, Jagath Samarabandu, *"Multiscale Edge-Based Text Extraction from Complex Images"*, International Conference on Multimedia and Expo, pp. 1721-1724, 2006.

[3] B. Bai, F. Yin, and C. L. Liu, *"Scene text localization using gradient local correlation.,* In Proc. ICDAR 2013, pp. 1380-1384, 2013.

[4] 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 Proc. ICIP 2011, pp. 2609-2612, 2011.

[5] P. Clark and M. Mirmehdi, *"Recognising text in real scenes,"* Int. Jour. on Document Analysis and Recognition, vol. 4, no. 4, pp. 243257, 2002.

[6] J. Ohya, A. Shio, and S. Akamatsu, *"Recognizing characters in scene images,"* IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 16, no. 2, pp. 214220, Feb. 1994.

[7] J. Matas, O. Chum, M. Urban, and T. Pajdla, *"Robust wide baseline stereo from maximally stable extremal regions,"* in Proc. BMVC 2002, pp. 384-396, 2002.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *"ImageNet classification with deep convolutional neural network,"* in Proc. NIPS, 2012, pp. 1097-1105.

[9] K. Kavukcuoglu, P. Sermanet, Y. Boureau, K. Gregor, M. Mathieu, and Y. LeCun, *"Learning convolutional feature hierarchies for visual recognition,"* in Proc. NIPS, 2010, pp. 1090-1098.

[10] Hojin Cho, Myungchul Sung, Bongjin Jun, *"Canny Text Detector: Fast and Robust Scene Text Localization Algorithm,"* 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 3566-3573.

[11] Y. Freund and R. E. Schapire, *"A decision-theoretic generalization of on-line learning and an application to boosting,"* Journal of Computer and System Sciences, 55(1) pp. 119139, 1997.

[12] International Conference on Document Analysis and Recognition (ICDAR 2015), Incidental Scene Text Dataset *"http://rrc.cvc.uab.es/?ch=4&com=downloads"*

[13] MathWorks, Learning and Fine-Tuning of Convolutional Neural Networks, *"https://in.mathworks.com/help/nnet/convolutional-neural-networks.html"*