Robust Reading in Scene Images

Aroma Mahendru 10105EN009
Mukund Laddha 10105EN036
Sudeep Soni 10105EN049
Manish K Gupta 10105EN052

Department of Electronics Engineering
Indian Institute of Technology (BHU) Varanasi

May 5, 2014

Supervised by:
Dr. R.R. Das
Professor
Department of Electronics Engineering
Indian Institute of Technology (BHU) Varanasi

Acknowledgements

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without mention of the people who made it possible and the support that had been a constant source of encouragement which in the end crowned our efforts with success.

We are deeply indebted and would like to express our sincere thanks to our Mentor and Guide Dr. R.R. Das, Professor, Department of Electronics Engineering IIT BHU, for providing us an opportunity to do this project under his guidance, constant encouragement and wholehearted support.

Our special gratitude to Dr. P.K. Jain , Head Department of Electronics Engineering, IIT BHU for his kind co-operation consistent motivation for research.

We also appreciate and thank our colleagues and juniors who have willingly helped us out with the best of their abilities.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Reading from scene images

Text in images contains valuable information and is exploited in many content-based image and video applications, such as content-based web image search, video information retrieval, mobile based text analysis and recognition. Due to complex background, variations of font, size, color and orientation, text in natural scene images has to be robustly detected before being recognized or retrieved.
A number of factors introduce challenges to the task of recognizing characters in natural scenes. These are:

1. Clutter and placement: where exactly in a natural scene is the text and how much of the scene is not relevant to character recognition?

2. Different font styles: outlines vs. solid, thick vs. thin lines, colours and textures, etc.

3. Variation in lighting conditions.

Figure 1.1: Examples of characters in natural scenes

1.2 Reading in Scene Images vs. OCR

In contrast to more classical OCR problems, where the characters are typically monotone on fixed backgrounds, character recognition in scene images is potentially far more complicated due to the many possible variations in background, lighting, texture and font. As a result, building complete systems for these scenarios requires us to invent representations that account for all of these types of variations. Indeed, significant effort has gone into creating such systems, with top performers integrating dozens of cleverly combined features and processing stages .Recent work in machine learning, however, has sought to create algorithms that can learn higher level representations of data automatically for many tasks. Such systems might be particularly valuable where specialized features are needed but not easily created by hand. Another potential strength of these approaches is that we can easily generate large numbers of features that enable higher performance to be achieved by classification algorithms.

1.3 Components of a Reading System

To achieve the above mentioned task we have divided our project into 3 systems:

1. Text Localisation and Verification: This system specifies the possible regions of characters in the image. As some region may be falsely detected due to background noise so we need to verify and remove them as well.

2. Letter Candidate Selection:After selecting text regions, we need a system to extract alphabet from text regions and send them to text detection system.

3. Text Recognition: This step covers the character recognition part of the problem like OCR except the fact that characters here are of large variety of fonts, colors and sizes.

In our project, we have tried to design and optimize 1. and 3. systems. For letter candidate selection, Sliding Window method is widely used and there are highly optimized programs available. Hence can be easily exploited.

1.4 Datasets

For the first system, we have used text localization dataset for *Robust Reading Competition Challenge 2 of ICDAR'13*.
For the third system, we have used *The Chars74K* dataset for character recognition in scene images. It comprises of three datasets :

1. characters obtained from natural images
2. hand drawn characters using a tablet PC
3. synthesised characters from computer fonts

We have used only the first data set as we focus on natural scenes spanning the 62 classes of characters (0-9, A-Z and a-z).

Chapter 2

Text Localization and Verification

Existing methods for scene text localization can roughly be categorized into two groups: machine learning based methods and image processing based methods.We have chosen the latter due to the fact that image processing methods are relatively computationally inexpensive and since they are based on the inherent properties of the text, the accuracy would not vary with the dataset. Moreover in image processing approach also there are a variety of approaches mainly based on two methods: gradient based, the approach using stroke width transform and wavelet transform based. We tested our dataset with first two and decided to finally use gradient based method because of more accuracy.

2.1 Basic methodology

Gradient based approach for video frames as proposed by A.Dutta et.all is very general. We have slightly modified their methodology in the last steps of the flow chart.

The natural scene images often suffer in degradation during transmission through various media, text portions can always be distinguished due to its discriminative pixel values with respect to the background. Text portions in an image always have distinct intensity values with respect to its background. The differences in the pixel values of an image are noted in the gradient of that image. Based on this observation our proposed method performs using gradient information and text edge map selection. We describe the entire implemented approach subsequently in the order where sub-section 2.2 describes gradient based procedure; edge based procedure in 2.3 and

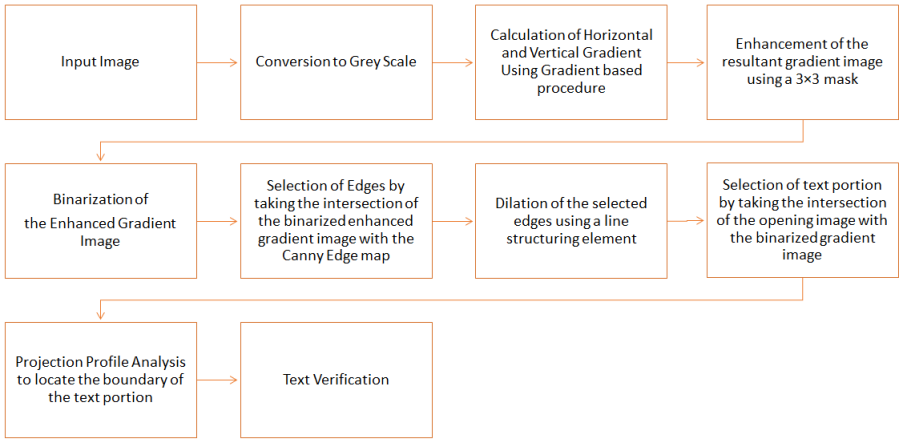

Figure 2.1: Basic Methodology flow chart

projection profile analysis in 2.4. Text box extraction and false positive removal are discussed in 2.5.

2.2 Gradient based procedure

In this method we first create horizontal and vertical gradient image of the original image and then merge them to get the resultant gradient image. To get the horizontal gradient image (HGI) of the original image (I) we consider the corresponding gray image (GI) of that image. The gradient value of a particular pixel (i, j) of HGI is calculated by taking the difference of the immediate lower pixel (i+1, j) with the current pixel (i, j) of the gray image (GI).

$$HGI(i,j) = \mid GI(i,j) - GI(i+1,j) \mid \forall (i,j) \in GI and (i+1,j) \in GI \quad (2.1)$$

Similarly the gradient value of a particular pixel (i, j) of the vertical gradient image (VGI) of the image (I) is calculated by taking the difference of the immediate right pixel (i, j+1) with the current pixel (i, j) of the gray imag in the following way:

$$VGI(i,j) = \mid GI(i,j) - GI(i,j+1) \mid \forall (i,j) \in GI and (i,j+1) \in GI \quad (2.2)$$

The horizontal and vertical gradient images are then merged to find the resultant gradient image (RGI):

$$RGI(i,j) = HGI(i,j) + VGI(i,j) \forall (i,j) \in HGI \bigcap VGI \quad (2.3)$$

Figure 2.2: (a) Original image (b) Gradient Image

Figure 2.3: Edge Detection

We enhance *RGI* using a 5x5 mask before further processing.

2.3 Edge based procedure

In this stage we first apply Canny edge detector to identify the edges in the image. This image is dilated. We then select edges of the image by taking the intersection of the binarized information of the enhanced gradient image (which we get from section 2.2) with the edge map.

2.4 Projection Profile Analysis

We next perform projection profile analysis (horizontal projection followed by vertical projection) to determine the boundary of the text region. This removes most of the noise edges in the image because horizontal lines containing text have more pixels with higher intensity values. An appropriate threshold is set for finding text rows.

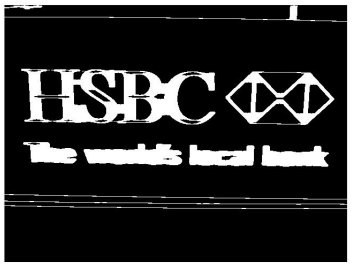

Figure 2.4: Projection Profile Image

Post processing

After these three steps, post processing is done including small size candidates are removed, hole removal and image closing.Then we send the resultant image to finalize text boxes.

2.5 Textbox Verification

Before finally plotting textboxes, we measure the aspect ratio and area ratio of light to dark pixels in the image. These two factors are very important in characterizing textboxes. By experimenting with the dataset, we have learned the aspect ratio and area ratio ranges. Then we have used it to separate identify and reject false textboxes.

Letter Candidate Selection

We found that we cannot use projection profile analysis just like in identifying text boxes for letter candidate selection. This is because the variance of vertical projection profile was found to be non-uniform and we have to consider each possibility.
Hence, usual sliding window method can be used to generate text candidates which can be sent to the classifier. Since, we are detecting single line texts, sliding window tool is only required to adjust the width without changing the height. This will save much processing time.
This also enables a possibility of confirmation in recognition. The letter candidate will be labelled to that alphabet which is the coincident result of two or more consecutive text boxes.
This can be pursued in future as an extension of the project.

Chapter 3

Text Recognition

Final and major step in our project is identifying alphabets in the letter candidates sent to the recognition system. We have done this by using Bag of Visual words model for finding uniform image features and trained a single layered neural network using back propagation algorithm.

3.1 Feature Extraction and Representation

3.1.1 Feature Extraction using SIFT features

Matching features across different images is a common problem in computer vision. When all images are similar in nature simple corner detectors can work. But when we have images of different scales and rotations, we need to use such features which are invariant in terms of scale, rotation, illumination and view point. There are lots of popular features like this including Scale Invariant Feature Transform (SIFT), Histogram of Gradient (HOG), Geometric Blur (GB), Spin Image (SI), Maximum Response of Filters (MR8), Patch Descriptors (PCH) etc.

We have used SIFT features in our project, which is one of the most widely used image descriptors these days.SIFT is quite an involved algorithm. Heres an outline of steps involved in SIFT algorithm:

1. Constructing a scale space
   This is the initial preparation. It creates internal representations of the original image to ensure scale invariance. This is done by generating a scale space.

2. LoG Approximation
   The Laplacian of Gaussian is great for finding interesting points (or

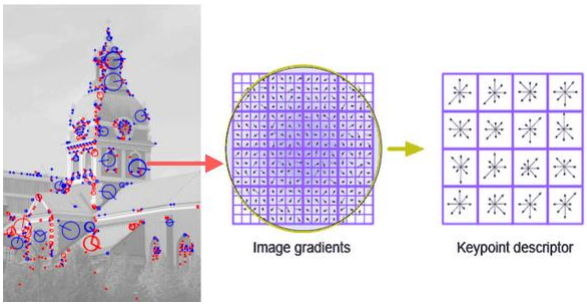

Figure 3.1: Extracting SIFT descriptors

key points) in an image. But its computationally expensive. So it approximates it using the representation created int the previous step.

3. Finding keypoints
   With the super fast approximation, it now tries to find key points. These are maxima and minima in the Difference of Gaussian image calculated in step 2.

4. Get rid of bad key points
   Edges and low contrast regions are bad keypoints. Eliminating these makes the algorithm efficient and robust. A technique similar to the Harris Corner Detector is used here.

5. Assigning an orientation to the keypoints
   An orientation is calculated for each key point. Any further calculations are done relative to this orientation. This effectively cancels out the effect of orientation, making it rotation invariant.

6. Generate SIFT features
   Finally, with scale and rotation invariance in place, one more representation is generated. This helps uniquely identify features.

We thus use a SIFT detector to find key points and their unique descriptor in each of our candidate image.

3.1.2 Bag of Visual Words model

Bag-of-visual-words is a popular technique for representing image content for object category recognition. The idea is to represent objects as histograms of feature counts. This representation quantizes the continuous

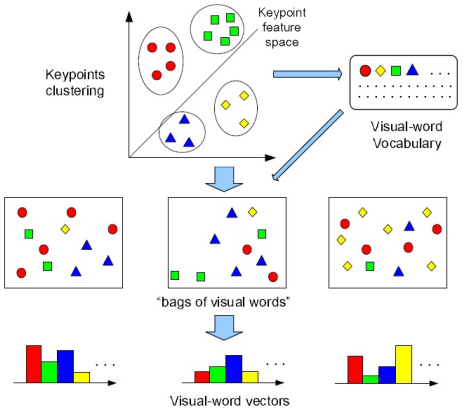

Figure 3.2: Bag of visual words method

high-dimensional space of image features to a manageable vocabulary of visual words. This is achieved, by grouping the low-level features collected from an image corpus into a specified number of clusters using an unsupervised algorithm K-Means. One can then map each feature extracted from an image onto its closest visual word and represent the image by a histogram over the vocabulary of visual words. We learn a set of visual words per class and aggregate them across classes to form the vocabulary. We learned 5, 10 and 15 visual words per class in our experiments. We decided with 10 words per class as it gave the maximum accuracy leading to a vocabulary of size more or less 620 words.
Hence each image is represented by a fixed size vector having around 620 elements each.

3.2 Classification using Neural Networks

Since we had to classify in 62 classes, classification was done with the help of artificial neural network. It was chosen because the classifications by Linear SVM and other similar algorithm can't generate complicated higher order functions required by us.
Due to memory issues, we trained a neural network with single hidden layer containing 300 nodes via back propagation algorithm.

Chapter 4

Implementation Results

4.1 Text Localization

We have tested our method on ICDAR '11 Robust Reading dataset and compared the results with the top entries of the event. Comparison is shown in the Table 4.1.

Table 4.1: ICDAR'11 Text Localization Results (%)

Method	*Recall*	*Precision*
Kims Method	62.47	82.98
Yis Method	58.09	67.22
TH-TextLoc System	57.68	66.97
Neumanns Method	52.54	68.93
TDM IACS	53.52	63.52
LIP6-Retin	50.07	62.97
KAIST AIPR System	44.57	59.67
ECNU-CCG Method	38.32	35.01
Text Hunter	25.96	50.05
Our Method	**72.25**	**60.30**

As we can see the performance of our gradient and edge based approach has very high recall and fairly high precision. The additional benefit is that since no learning is being done, we are saving processing power.

4.2 Text Recognition

As mentioned before, we tried with different vocabulary sizes to find the maximum accuracy. Results are shown in Table 4.2.

Table 4.2: Accuracy variation with vocabulary size

Number of words per class	*Training set Accuracy(%)*
5	41.45
10	46.22
15	44.60

After this we divided our dataset into training and test set in 4:1 ratio. Number of words per class used was 10. We used training set for training the net and test set for comparing the results.Accuracies are shown in Table 4.3:

Table 4.3: Training, Test Accuracy

Dataset	*Accuracy(%)*
Training Set (80%)	41.89
Test Set (20 %)	38.26

Since the dip in test set accuracy is not very large, it can be said that the behavior of the classifier would not deviate much from the range. Furthermore, commercial softwares like ABBYY have accuracies in the range of 60-65 percent on similar dataset. Thus our accuracy with a single hidden layer neural network classifier is fairly high.

4.3 Conclusion and Future Work

To conclude, text localization by gradient based approach is very affective in in recognizing text boxes but the rate of false alarm due to noise is a bit high too. Using neural networks to classify bag of visual words data of character is image is efficient but inconsistency in number of features induces a lot of errors.
Extension of this work , according to us should include usage of sliding window candidates in letter confirmation, using more accurately characterizing

features like Geometric blur descriptors or spin count features and may be trying more advanced machine learning algorithm like multiple kernel learning and convolutional neural networks.

Appendix

Full implementation of the Robust Reading system developed will be available at: https://github.com/aroma123/rr
The bag of words model and histogram generation is based on the practical image classification code by A. Vedaldi and A. Zisserman, Robotics group, Oxford University UK. This code is available at:
http://www.robots.ox.ac.uk/ vgg/share/practical-image-classification.htm

References

Shahab A., Shafait F. and Dengel A., *ICDAR 2011 Robust Reading Competition Challenge 2: Reading Text in Scene Images*, International Conference on Document Analysis and Recognition (ICDAR), 2011 , pp.1491-1496, 18-21 Sept. 2011

Campos T. E., Babu B. R., Varma M., *Character Recognition in Natural Images.*VISAPP (2) 2009: 273-280

Ye Q., Gao W., Wang, W. , Zeng W., *A robust text detection algorithm in images and video frames.* Joint Conference of Fourth International Conference on Information Communications and Signal Processing and Pacific-Rim Conference on Multimedia, Singapore.

Coates A., Carpenter B., Case C., Satheesh S., Suresh B., Wang T., Wu D., Ng A. :*Text Detection and Character Recognition in Scene Images with Unsupervised Feature Learning*,ICDAR 2011

Dutta A.,Pal U., Shivakumara P., Ganguli A., Bandyopadhya A., Tan C. L.: *Gradient based Approach for Text Detection in Video Frames*, ICSIP2009, pp 387-393.

Vedaldi A. and Fulkerson B.,*VLFeat*: An Open and Portable Library of Computer Vision Algorithms, http://www.vlfeat.org, 2008.