Automatic text detection

Alkesh (14BCE0658),Kanuj (14BCE0662),Anirudh (14BCE0602)

November 10, 2016

1 Abstract

Large amounts of information are embedded in natural scenes.Signs are good examples of natural objects with high information content. In this project, we will discuss problems in automatic detection and translation of text from natural scenes. We describe the challenges of automatic text detection and propose methods to address these challenges. We extend example based machine translation technology for sign translation and present a prototype-system for sign board translation. This system is capable ofcapturing images, automatically detecting and recognizing text,and translating the text . The translation can be displayed on a palm size PDA, or synthesized as a voice output.Text recognition in images is a research area which attempts to develop a computer system with the ability to automatically read the text from images. These days there is a huge demand in storing the information available in paper documents format in to a computer storage disk and then later reusing this information by searching process. One simple way to store information from these paper documents in to computer system is to first scan the documents and then store them as images. But to reuse this information it is very difficult to read the individual contents and searching the contents form these documents line-by-line and word-by-word. The challenges involved in this the font characteristics of the characters in paper documents and quality of images. Due to these challenges, computer is unable to recognize the characters while reading them. Thus there is a need of character recognition mechanisms to perform Document Image Analysis (DIA) which transforms documents in paper format to electronic format. In this paper we have reviewed and analyzed different methods for text recognition from images. The

objective of this review paper is to summarize the well-known methods for better understanding of the reader

2 Introduction

Reading text from photographs is a challenging problem that has received a significant amount of attention. Two key components of most systems are (i) text detection from images and (ii) character recognition, and many recent methods have been proposed to design better feature representations and models for both. In this paper, we apply methods recently developed in machine learningspecifically, large-scale algorithms for learning the features automatically from unlabeled dataand show that they allow us to construct highly effective classifiers for both detection and recognition to be used in a high accuracy end-to-end system.Signs are everywhere in our lives. They suggest the presence of a fact, condition, or quality. They make our lives easier when we are familiar with them, but sometimes they pose problems or even danger. For example, a tourist or soldier might not be able tounderstand a sign in a foreign country that specifies militarywarnings or hazards. These signs include street and company names, bulletin boards, announcements, advertisements, and warning notices, and others. Information is scanned through paper documents as we know that we have number of newspapers and books which are in printed format related to different subjects. These days there is a huge demand in storing the information available in these paper documents in to a computer storage disk and then later reusing this information by searching process. One simple way to store information in these paper documents in to computer system is to first scan the documents. Whenever we scan the documents through the scanner, the documents are stored as images format in the computer system. These images containing text cannot be edited by the user. But to reuse this information it is very difficult for computer system to read the individual contents and searching the contents form these documents line-by-line and wordby-word. The reason for this difficulty is the font characteristics of the characters in paper documents are different to font of the characters in computer system. As a result, computer is unable to recognize the characters while reading them. Reading text from photographs is a challenging problem that has received a significant amount of attention. Two key components of most systems are (i) text detection from images and (ii) character recognition, and many recent

methods have been proposed to design better feature representations and models for both.

3 Theorical Analysis

Sign translation, in conjunction with spoken language translation, can help international tourists to overcome language barriers. A successful sign translation system relies on three key technologies: text detection, optical character recognition (OCR), and language translation. At current stage of the research, we focus our efforts on automatic text detection and sign translation while taking advantage of existing OCR technologies.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 . Extremal regions have two important properties, that the set is closed under: 1. continuous transformation of image coordinates. This means it is affine invariant and it doesn't matter if the image is warped or skewed. 2. monotonic transformation of image intensities. The approach is of course sensitive to natural lighting effects as change of day light or moving shadows. Optical character recognition (OCR) is the mechanical or electronic conversion of images of typed, handwritten or printed text into machine-encoded or from subtitle text superimposed on an image. It is widely used as a form of information entry from printed paper data records, whether passport documents, invoices, bank statements, computerised receipts, business cards, mail, prinouts of static-data, or any suitable documentation. In computer vision, maximally stable extremal regions (MSER) are used as a method of blob detection in images. This technique was proposed by Matas et al. To find correspondences between image elements from two images with different viewpoints. This method of extracting a comprehensive number of corresponding image elements contributes to the wide-baseline matching, and it has led to better stereo matching and object recognition algorithms. Because the regions are defined exclusively by the intensity function in the region and the outer border, this leads to many key characteristics of the

regions which make them useful. Over a large range of thresholds, the local binarization is stable in certain regions, and have the properties listed below. Invariance to affine transformation of image intensities Stability: only regions whose support is nearly the same over a range of thresholds is selected. Multi-scale detection without any smoothing involved, both fine and large structure is detected. Note, however, that detection of MSERs in a scale pyramid improves repeatability, and number of correspondences across scale changes. The set of all extremal regions can be enumerated in worst-case, where is the number of pixels in the image. Automatic detection of signs from natural scenes is a challenging problem because they usually embedded in the environment. The task is related to text detection and recognition from video images, or so called Video OCR. Compared to video OCR tasks, sign detection takes place in a more dynamic environment. The users movement can cause unstable input images. Non-professional equipment can make the video input poorer than that of other video OCR tasks, such as detecting captions in broadcast news programs. Sign detection must also be implemented in real time using limited resources. So in this project we will try to recognize text and this can be implemented in language translation from one language to another.

4 Experimentation

Step 1: Detect Candidate Text Regions Using MSER The MSER feature detector works well for finding text regions . It works well for text because the consistent color and high contrast of text leads to stable intensity profiles.The detectMSERFeatures function is used to find all the regions within the image and plot these results.There may be many non text region alongside of the image.

Step 2: Remove Non-Text Regions Based On Basic Geometric Properties Although the MSER algorithm picks out most of the text, it also detects many other stable regions in the image that are not text. We can use a rule-based approach to remove non-text regions. For example, geometric properties of text can be used to filter out non-text regions using simple thresholds. Alternatively, we can use a machine learning approach to train a text vs. non-text classifier. Typically, a combination of the two approaches produces better results . This example uses a simple rule-based approach to filter non-text regions based on geometric properties. Use regionprops to

measure a few of these properties and then remove regions based on their property values.

Step 3: Remove Non-Text Regions Based On Stroke Width Variation Another common metric used to discriminate between text and non-text is stroke width. Stroke width is a measure of the width of the curves and lines that make up a character. Text regions tend to have little stroke width variation, whereas non-text regions tend to have larger variations.

Step 4: Merge Text Regions For Final Detection Result At this point, all the detection results are composed of individual text characters. To use these results for recognition tasks, such as OCR, the individual text characters must be merged into words or text lines. This enables recognition of the actual words in an image, which carry more meaningful information than just the individual characters. For example, recognizing the string 'EXIT' vs. the set of individual characters 'X','E','T','I', where the meaning of the word is lost without the correct ordering. One approach for merging individual text regions into words or text lines is to first find neighboring text regions and then form a bounding box around these regions. To find neighboring regions, expand the bounding boxes computed earlier with regionprops. This makes the bounding boxes of neighboring text regions overlap such that text regions that are part of the same word or text line form a chain of overlapping bounding boxes.

Step 5: Recognize Detected Text Using OCR After detecting the text regions, use the ocr function to recognize the text within each bounding box. Note that without first finding the text regions, the output of the ocr function would be considerably more noisy.

4.1 Program

```
colorImage = imread('sign_board.jpg');
I = rgb2gray(colorImage);

% Detect MSER regions.
[mserRegions, mserConnComp] = detectMSERFeatures(I, ...
    'RegionAreaRange',[200 8000],'ThresholdDelta',4);

figure
imshow(I)
hold on
```

```
plot(mserRegions, 'showPixelList', true,'showEllipses',false)
title('MSER regions')
hold off
% Use regionprops to measure MSER properties
mserStats = regionprops(mserConnComp, 'BoundingBox', 'Eccentricity', ...
    'Solidity', 'Extent', 'Euler', 'Image');

% Compute the aspect ratio using bounding box data.
bbox = vertcat(mserStats.BoundingBox);
w = bbox(:,3);
h = bbox(:,4);
aspectRatio = w./h;

% Threshold the data to determine which regions to remove. These thresholds
% may need to be tuned for other images.
filterIdx = aspectRatio' > 3;
filterIdx = filterIdx | [mserStats.Eccentricity] > .995 ;
filterIdx = filterIdx | [mserStats.Solidity] < .3;
filterIdx = filterIdx | [mserStats.Extent] < 0.2 | [mserStats.Extent] > 0.9;
filterIdx = filterIdx | [mserStats.EulerNumber] < -4;

% Remove regions
mserStats(filterIdx) = [];
mserRegions(filterIdx) = [];

% Show remaining regions
figure
imshow(I)
hold on
plot(mserRegions, 'showPixelList', true,'showEllipses',false)
title('After Removing Non-Text Regions Based On Geometric Properties')
hold off

% Compute the stroke width variation metric
strokeWidthValues = distanceImage(skeletonImage);
strokeWidthMetric = std(strokeWidthValues)/mean(strokeWidthValues);
% Threshold the stroke width variation metric
strokeWidthThreshold = 0.4;
```

```
strokeWidthFilterIdx = strokeWidthMetric > strokeWidthThreshold;
% Process the remaining regions
for j = 1:numel(mserStats)

    regionImage = mserStats(j).Image;
    regionImage = padarray(regionImage, [1 1], 0);

    distanceImage = bwdist(~regionImage);
    skeletonImage = bwmorph(regionImage, 'thin', inf);

    strokeWidthValues = distanceImage(skeletonImage);

    strokeWidthMetric = std(strokeWidthValues)/mean(strokeWidthValues);

    strokeWidthFilterIdx(j) = strokeWidthMetric > strokeWidthThreshold;

end

% Remove regions based on the stroke width variation
mserRegions(strokeWidthFilterIdx) = [];
mserStats(strokeWidthFilterIdx) = [];

% Show remaining regions
figure
imshow(I)
hold on
plot(mserRegions, 'showPixelList', true,'showEllipses',false)
title('After Removing Non-Text Regions Based On Stroke Width Variation')
hold off
% Get bounding boxes for all the regions
bboxes = vertcat(mserStats.BoundingBox);

% Convert from the [x y width height] bounding box format to the [xmin ymin
% xmax ymax] format for convenience.
xmin = bboxes(:,1);
ymin = bboxes(:,2);
xmax = xmin + bboxes(:,3) - 1;
ymax = ymin + bboxes(:,4) - 1;
```

```
% Expand the bounding boxes by a small amount.
expansionAmount = 0.02;
xmin = (1-expansionAmount) * xmin;
ymin = (1-expansionAmount) * ymin;
xmax = (1+expansionAmount) * xmax;
ymax = (1+expansionAmount) * ymax;

% Clip the bounding boxes to be within the image bounds
xmin = max(xmin, 1);
ymin = max(ymin, 1);
xmax = min(xmax, size(I,2));
ymax = min(ymax, size(I,1));

% Show the expanded bounding boxes
expandedBBoxes = [xmin ymin xmax-xmin+1 ymax-ymin+1];
IExpandedBBoxes = insertShape(colorImage,'Rectangle',expandedBBoxes,'LineWidth'

figure
imshow(IExpandedBBoxes)
title('Expanded Bounding Boxes Text')
% Compute the overlap ratio
overlapRatio = bboxOverlapRatio(expandedBBoxes, expandedBBoxes);

% Set the overlap ratio between a bounding box and itself to zero to
% simplify the graph representation.
n = size(overlapRatio,1);
overlapRatio(1:n+1:n^2) = 0;

% Create the graph
g = graph(overlapRatio);

% Find the connected text regions within the graph
componentIndices = conncomp(g);
% Merge the boxes based on the minimum and maximum dimensions.
xmin = accumarray(componentIndices', xmin, [], @min);
ymin = accumarray(componentIndices', ymin, [], @min);
xmax = accumarray(componentIndices', xmax, [], @max);
```

```
ymax = accumarray(componentIndices', ymax, [], @max);

% Compose the merged bounding boxes using the [x y width height] format.
textBBoxes = [xmin ymin xmax-xmin+1 ymax-ymin+1];
% Remove bounding boxes that only contain one text region
numRegionsInGroup = histcounts(componentIndices);
textBBoxes(numRegionsInGroup == 1, :) = [];

% Show the final text detection result.
ITextRegion = insertShape(colorImage, 'Rectangle', textBBoxes,'LineWidth',3);

figure
imshow(ITextRegion)
title('Detected Text')
ocrtxt = ocr(I, textBBoxes);
[ocrtxt.Text]
```

5 Conclusion

This example showed you how to detect text in an image using the MSER feature detector to first find candidate text regions, and then it described how to use geometric measurements to remove all the non-text regions. This example code is a good starting point for developing more robust text detection algorithms.

6 Acknowledgment

We would like extend our sincere gratitude to our Image and Vision Computing course faculty , Prof. Don S who encouraged us to do this project and to explore the domain of Digital Image Processing and Computer Vision.

7 Reference

Brown, R.D., Example-based machine translation in the pangloss system.Proceedings of the 16th International Conference on Computational Linguistics

Cui, Y. and Huang, Q., Character Extraction of License Plates from Video. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition

Jie Yang, Xilin Chen, Jing Zhang, Ying Zhang, Alex Waibel AUTOMATIC DETECTION AND TRANSLATION OF TEXT FROM NATURAL SCENES

mathworks