Projet 3A

TEXT DETECTION AND RECOGNITION

February 21, 2016

Zhixing CAO, Yuesong SHEN
Ecole Polytechnique

Contents

1 Introduction

The development of smartphones and the growing demands in content-based image understanding has made the text detection a crucial topic in machine-human interaction. It has been shown that the performance of image retrieval depends critically on the performance of text detection and recognition. For example, two book covers with different titles but identical background prove to be considered virtually indistinguish without detecting and recognizing the text.

1.1 Goal of project

Early approach of text detection and recognition techniques such as OCR can be traced in early 1900s. Recent years, with the progress in the field of machine learning pattern recognition and text localisation techniques make new breakthroughs. With the study of some of those techniques, our project concern algorithms that can decode the text in images.

The approach of our project can be divided into two parts — text detection and text recognition. The detection part detect potentiel texts in images and output pure text candidates and the recognition part translate the image of text candidates to digital texts that can be read by machines.

1.2 Work breakdown

The whole project is carried out by Zhixing CAO and Yuesong SHEN. Several works have been carried out independently during the project. Here is the overview of work breakdown.

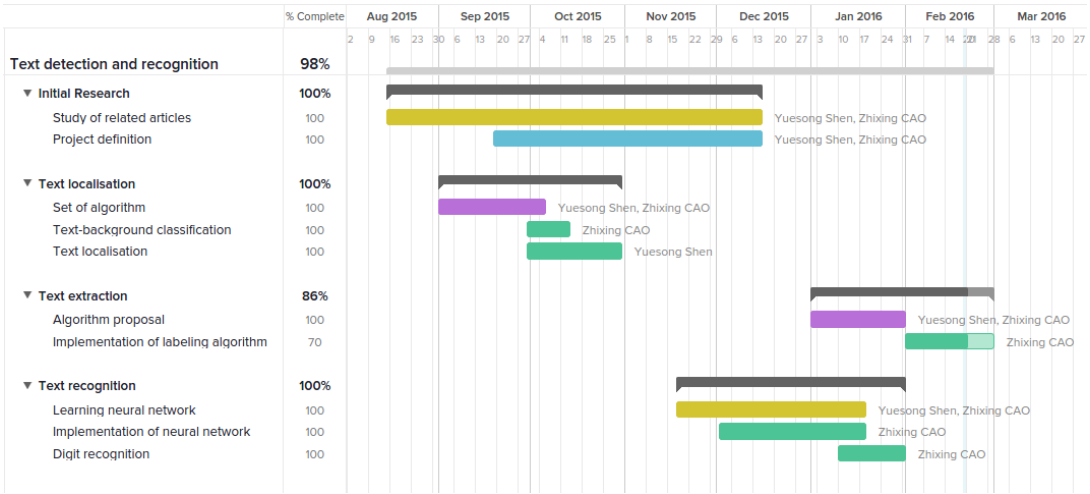

Figure 1: Work breakdown

2 State of the art

2.1 Text detection

As an essential prerequiste for text recognization, text within images has to be robustly located. This is challenge task due to the variety of the text form, such as variations in languages, font and style, geometric and photometric distortions, partial occlusion, and lightening conditions. Text detection problem have been studied a lot in recent researches and numerous methods are reported in the literature.

All the methods used in recent research can be classified into two catagories: method based on texture and method based on connected componet.

Texture-based method views texts as a special texture that is distinguish from its background. Features are extracted over some special regions and a classifier is used to identify texts areas. Zhong and al have segmented caption text regions from background and used the intensity variation information. Ye and al have proposed a method using multiscale wavelet featires and a coarse-to-fine algorithm to locate text lines under different backgrounds.

Different with the texture-based method, the connected component based approach extrats regions from the image and selects text candidates using some geometric rules. In ICDAR 2005 text locating competition, the best result applies have used an adaptive binarization method to find connected components and forms text lines based on geometric properties. Recently Chen and al extract letter candidates by employing edge-enhanced Maximally Stable Extremal Regions and using geometric and stroke width information to exclude non-text objects. They have achieved an accuracy score of 95%.

2.2 Text recognition

Converting text data from image and deciphering into digits is an important problem. Early physical photocell-based OCR implemented matrix matching by comparing an image to a stored glyph on a pixel-by-pixel based. Those algorithms involve mostly extensive processing on the image such as thinning, smoothing contour analysis etc. because the majority of previous works uses geometrial and topological features. In recent research community the dominant approach to this problem is based on machine learning techniques — a general inductive process builds automatically a classifier by learning.

In the textbook *Pattern Recognition and Machine Learning*, Bishop reflects recent developments in the field of pattern recognition and machine learning and shows potential usages of machine learning method in pattern recognition. The early effort has been made around 1986s by Burr, Mehr&Richfield to implement a neural networks in character recognition. Recently, with the development of paralel computation and the use of GPU, efficiant OCR system based on neural network is realized.

2.3 Our approach

In this report, we show you our approach of the text detection problem by combined texture-based method and connected component based method and our text recognition algorithm using neural network.

We adapte at first the method proposed by Liu and al to locate texts. This algorithm detects text candidate by applied classifier on contour pictures of the original image. Experimental results demonstrate that this approach is robust for font-size, font-color, background and languages, which can be used efficiently.

After locating the text, a connected component based method is used to extract word candidates. We find all components by using connected-component labeling algorithm. Then we use some geometric constraints and heuristic rules to merge them and extract letters and words candidates.

Our approach for text recognition is based on using neural network to classify characters. With a training set of different kind of characters, the neural network is constructed in order to match an input character to a learned one.

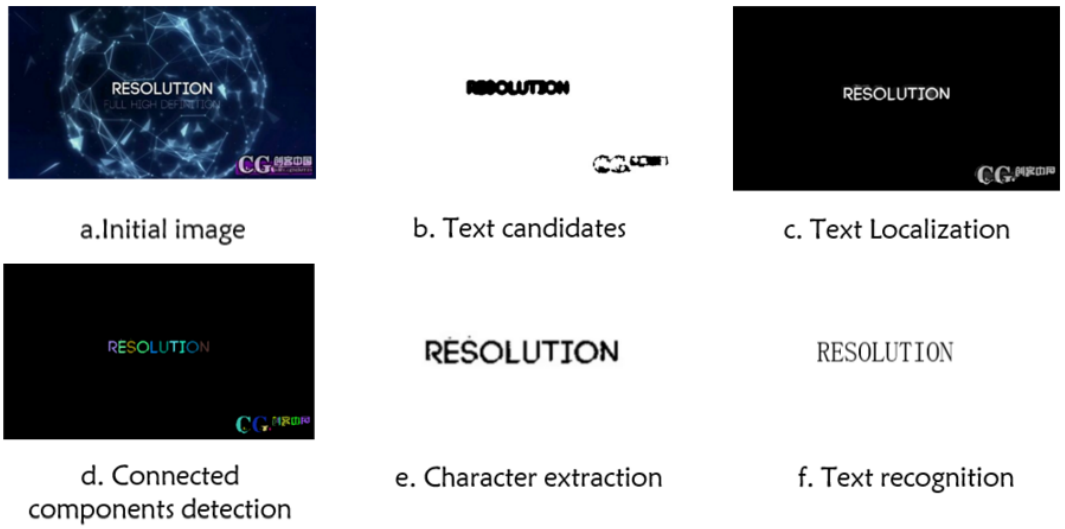

Figure 2: Overview

3 Text detection

In order to extract text from an image, we first need to determine the location of text zone in order to later perform the extraction and the recognition.

3.1 Detecton of contour

Since text is always composed of strokes (so that it can be written by men), edge turns out to be an excelent characteristic for text zone identification. We have adopted the approach proposed by Liu and al. for edge information extraction, that is by applying Sobel edge detector on the image for 4 directions (vertical, horizontal, up-right to down-left and up-left to down-right) with distance defined on the RGB color space to get 4 edge maps. And this 4 edge maps are to be used to determine the text locations.

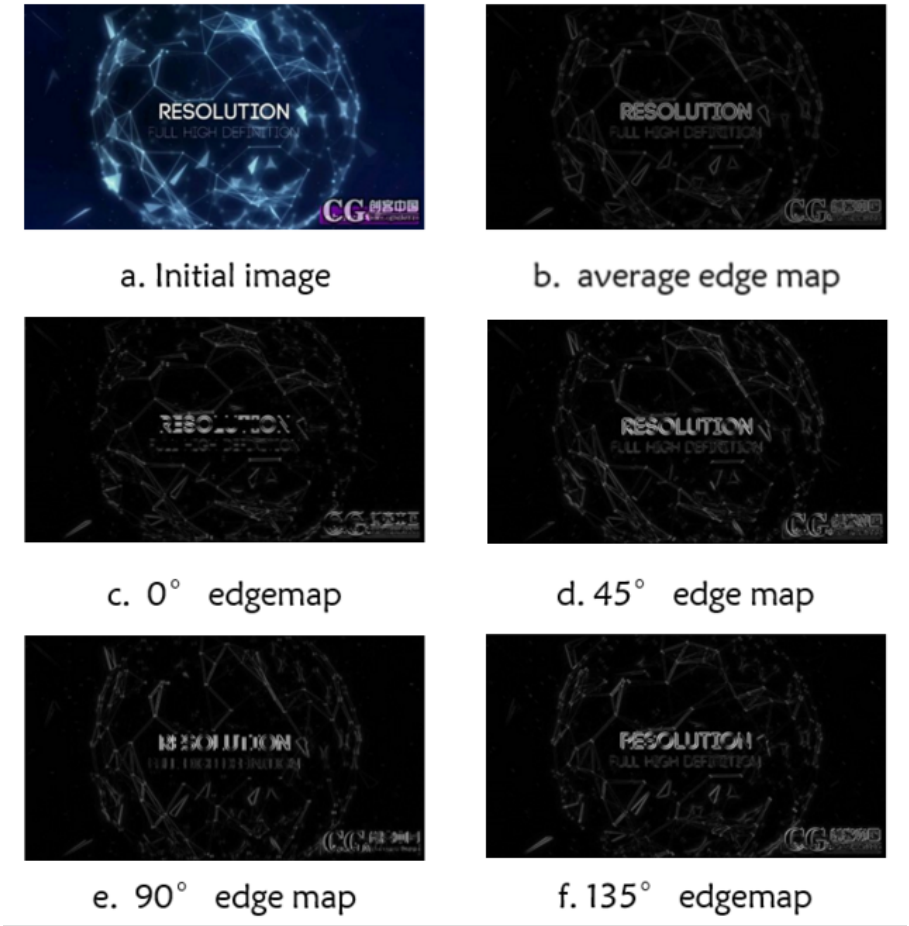

Figure 3: Edge maps

3.2 K-means classification

With the 4 edge maps, we can then apply a sliding window of size $w \times h$ to calculate the 6 different features as follows:

- $\mu = \frac{1}{w \times h} \sum_{i=1}^{w} \sum_{j=1}^{h} E(i, j)$

- $\sigma = \sqrt{\frac{1}{w \times h} \sum_{i=1}^{w} \sum_{j=1}^{h} (E(i,j) - \mu)}$
- $Et = \sum_{i,j} E(i,j) \log E(i,j)$
- $I = \sum_{i,j} (i-j)^2 E(i,j)$
- $H = \sum_{i,j} \frac{1}{1+(i-j)^2} E(i,j)$

Where $E(i,j)$ is the value of pixel in ith row, jth column, here it is the grayscale color of each pixel. μ is the mean value in sliding window, σ is their standard deviation, Et is their entropy, I is their gravity center and H is the expectation.

With the 4 edge maps, we then get 24 features for a given position of the sliding window. And since there is no apparent cost function available, we need an unsupervised clustering algorithm to distinguish text zones from their surroundings. We apply therefore the k-means algorithm for this, as proposed in the paper of Liu and al..

This approach, while effective for seperatings text and non-text zones, can not determine which one of the 2 zones is text zone, due to its non-supervised nature. We therefore need some extra effort to choose the text zone. There is no solution proposed in the original article. We propose 2 ideas to solve this problem: The first approach is to collect a set manually choosed text and non-text data (an average feature vector for each zone in each image) and then train a classifier with supervised learing approach. The second approach is to use geometrical and topological informations of each zone to determine text zones.

3.3 Text area identification

The result obtained by k-means algoritm needs to be polished to remove noise and other non text zone. We first use morphology operations âĂIJopenâĂİ and âĂIJdilateâĂİ to fill up tiny holes and gaps and remove too small zones in the background. We can then apply some empirical rules to further remove zones which can not contain text. Each connected component of the refined text zones will then be boxed by a rectangle and returned as final results.

Figure 4: Text area identification

3.4 Connected components detection

The former method shows efficient results. However, it is still incapable to distinguish each character which is important for text recognition. So we use at next step the connected connected components algorithm to extract characters.

At first, we set the input image into a binary matrix according to a threshold. Knowing that all colors can be encoded into a grayscale double number between 0 and 255, we can transform our input image by set all pixels to 0 or 1 with their grayscale number by a threshold. Most of the time, pixels in a single letter have similar colors, so that in the binary matrix, these pixels have always the same value which will be considered as a same component.

The connected-component labeling algorithm is based on union-find method. The first pass of the algorithm propagate a pixel's label to its eight neighbors. Whenever the situation of connectivity arises, we attribute labels and union two set if it's neccesary. At the end of the first pass, each equivalence class has been completely determined and has a unique label, which is the root of its tree in the union-find structure. A second pass through the image then performs a translation, assigning to each pixel the label of its equivalence class.

Figure 5: Connected components detection

3.5 Character extraction

In many language, a single character can be composed by several different part. In the character extraction, those different components should be considered as a same part. So after the connected component detection. We calculate the gravity center Gi of each component and use gravity-color distance to centering them.

Gravity-color distance of component i and j:

$\sqrt{(euclidiendistanceofGi,Gj)^2+(greyscaleofi-greyscaleofj)^2}$

What's more, as characters usually have regular forms, which is to say, the ratio between their width and their height are never too large or two small. We use the height and the width of each connected component to filter non-text cadidate as well.

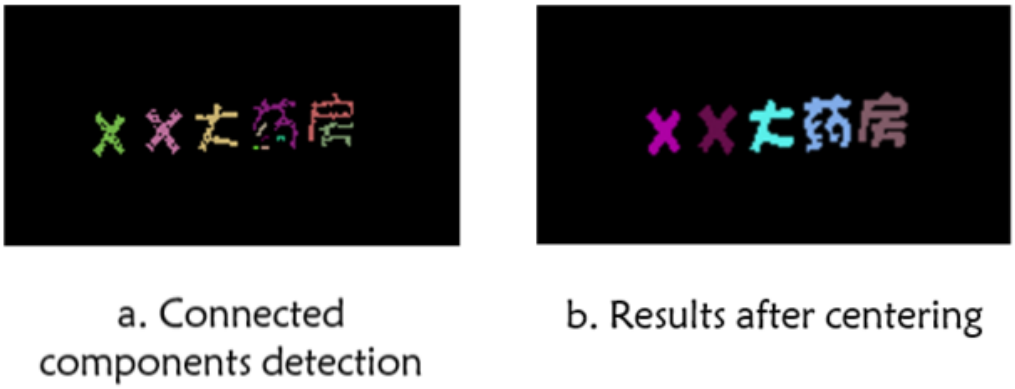

Figure 6: Components centering example

4 Text recognition

4.1 Neural network

After extract characters, we can move to the next step and try to recognize them.

In machine learning, neural network is a powerful tool to estimate functions that maps input to output. Theoretically, it can approximate every non-linear functions when using non-linear activation function.

Inspired of humans' central nervous systems, neural network uses connected nodes, known as neurons, to imitate the nervous system. A neural network is a complex adaptive system with ability to *learn*. It consists of multiple layers. Apart from the input layer, all layers have an activation function. By adjusting this function according to our given datas, known as training set, the network achieve to learn and understand these datas. Here is the structure of the neural network:

Our goal is to determine the activation function Θ by using gradient decent. At first, we need to set Θ to some random values in order to break the symmetry. Then, the forward propagation will determine the output while the backpropagation will correct the error.

In our project, we use neural network to learn digital numbers.

4.2 Pattern recognition

In essence, pattern recognition convers the problem:'Given examples of signals and the correct decisions of them, make decisions automatically with future streams'. In our project, we use the neural

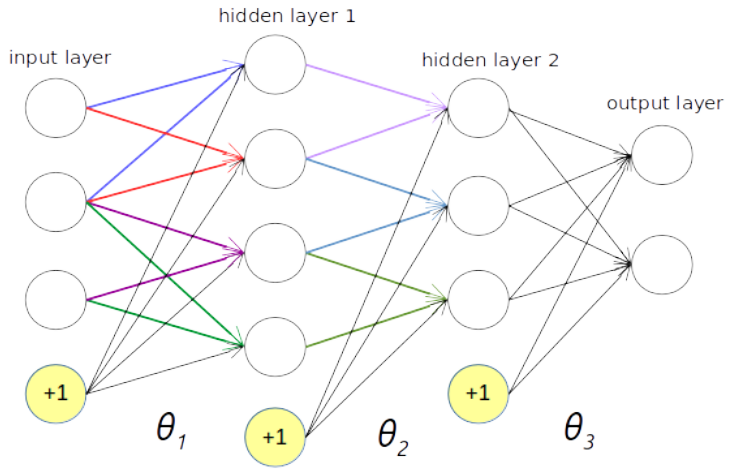

Figure 7: Structure of neural network

network to learn a set of examples of digit and letter images. Then we use the same neural network to predict new images.

For digital number recognition, we use MNIST dataset where all digit images have been size-normalized and centered in a fixed size image of 28×28 pixels. And for English letter recognition, we use the Chars74K dataset where each image has a size of 1200×900 pixels. For simplify our task, we resize the size of those images to 40×30. Therefore, each image of digit has 784 features and each image of letter has 1200 features.

Then we put all inputs into the neural network and using the trained neural network to recognize the extracted characters from the former step.

5 Result

5.1 Text detection

Here are the text detection result with different photos:

5.2 Text recognition

We show you here some results with digit recognition.

We use the MNIST database which consists of digits written by high school students and employees of the United States Census Bureau as our training data. The MNIST dataset consists of 70.000 handwritten digit immages of 0 to 9.

We have tested the network with 10.000 different images and with different parameters of the network.

Figure 8: Heatmap of confusion matrix

5.2.1 Influence of different parameters

Here are some results with different parameters:

num_training_data	*num_layers*	*num_nodes*	*num_iteration*	*accuracy*
60000	1	200	30	92.56
6000	1	200	30	91.96
6000	1	20	30	91.02
6000	1	10	30	84.59
6000	2	20,20	30	85.7
6000	2	20,20	45	90.33

We notice that we can get fairly good result by well choosing the parameters of the neural network such as the number of layers, number of nodes, size of training data etc..

5.2.2 Confusion matrix

In order to better visualize the performance of the algorithm, we use here the confusion matrix to see if the system is confusing two classes. Here is the heatmap of the confusion matrix:

Each color correspond with $log\frac{number\_of\_predicted\_i}{number\_actual\_i}$, with this map, we can easily determine the

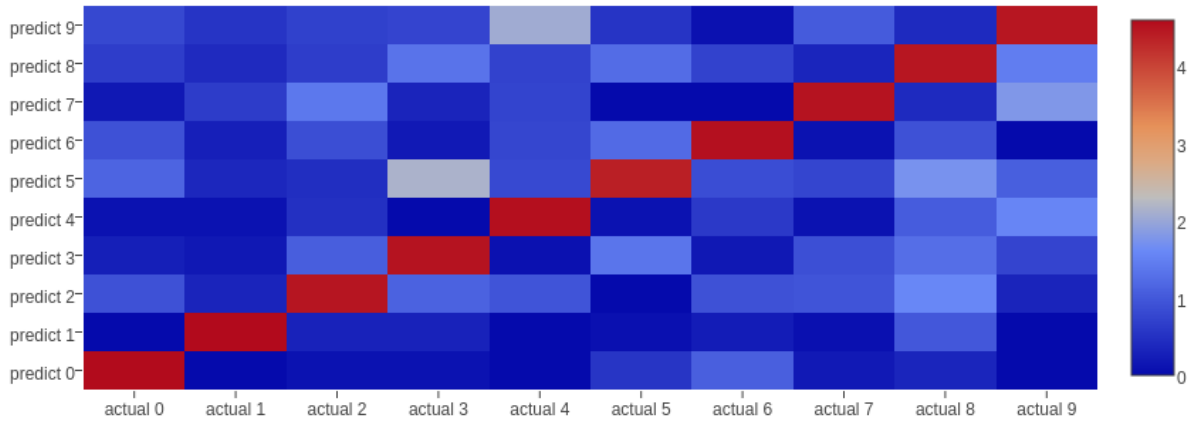

Figure 9: Heatmap of confusion matrix

error caused by the confusion of numbers. For exemple, the wrong predict of number 3 is always 5 and the wrong predict of number 4 is always 9. That is to say, the hand-writting of 4 is similary to the hand-writting of 9 and sometimes, it may lead to confusion to our system.

5.3 Remaining difficulties and possible improvement

6 Conclusion

References

Anand, U., 2010. The Elusive Free Radicals, *The Clinical Chemist,* [e-journal] Available at:<http://www.clinchem.org/content/56/10/1649.full.pdf> [Accessed 2 November 2013]

Biology Forums, 2012. *Normal glomerulus. Acute glomerulonephritis.* [online] Available at: <http://biology-forums.com/index.php?action=gallery;sa=view;id=9284> [Accessed 23 October 2013].