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 have 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 [5]. A machine can distinguish these two books only by recognize their titles.

## 1.1 Project definition

Early approach of text detection and recognition techniques such as OCR techniques 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 [1]. With the study of some of those techniques, our project concern algorithms that can decode text in images.

The approach of our project can be divided into two parts — text detection and text recognition. The detection part detect potentiel text in images and output text candidates and the recognition part translate the image of text candidates so that machine can understand.

## 1.2 Work breakdown

The whole project is carried out by Zhixing CAO and Yuesong SHEN. We use about one month reading articles and set our task into three independent parts: text area identification, character extraction and character recognition.

All sub-tasks require studies of related articles and implementation works. Yuesong SHEN is in charge of the text area identification part and Zhixing CAO is in charge of the character extraction and recognition part. Here is the overview of our schedule:

## 1.3 State-of-the-art

### 1.3.1 Text detection

As an essential prerequiste for text recognization, text within images has to be robustly located. This is a 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 has been studied a lot in recent researches and numerous methods are reported in the literature.

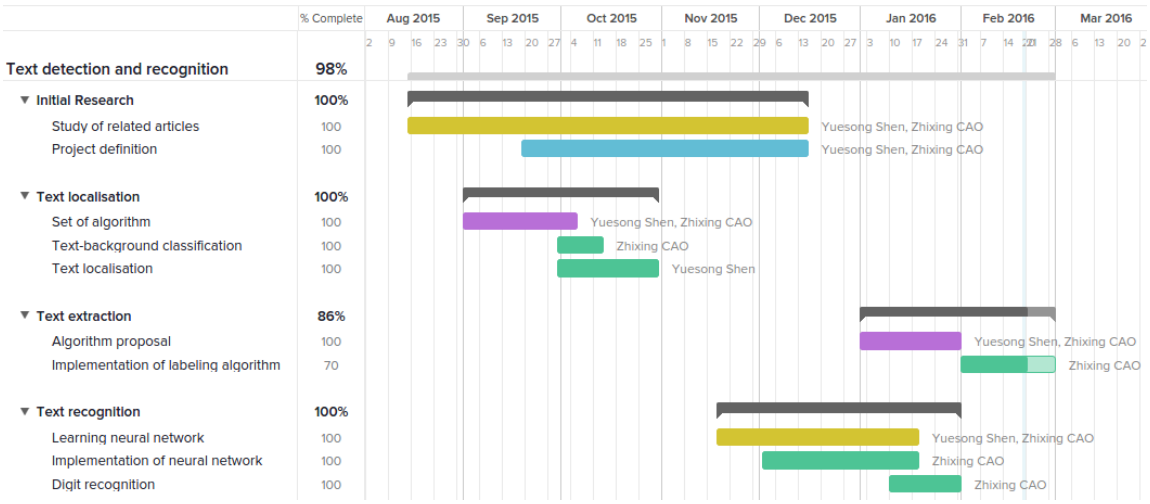



**Figure 1:** Schedule

All the methods used in recent research can be classified into two catagories: method based on texture [12] [7] and method based on connected componet [13] [6].

Texture-based method views texts as a special texture that is distinguishable from its background. Features are extracted over some special regions and a classifier is used to identify texts areas. Zhong et al. [14] have segmented caption text regions from background and used the intensity variation information. Ye and al [12] have proposed a method using multiscale wavelet features and a coarse-to-fine algorithm to locate text lines under different backgrounds.

Different with the texture-based method, the connected component based approach extracts regions from the image and selects text candidates using some geometric rules. In ICDAR 2005 text locating competition [10], the best result applies have used an adaptive binarization method to find connected components and forms text lines based on geometric properties. Recently Chen et al. [3] 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%.

### 1.3.2 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 basis. Those algorithms involve mostly extensive processing on the image such as thinning, smoothing contour analysis etc. because the majority of previous works uses geometrical and topological features. In recent research community the dominant approach to this problem is based on machine learning techniques — a general inductive process building automatically a

classifier by learning.

In the textbook *Pattern Recognition and Machine Learning* [1], 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 [2], Mehr& Richfield [11] to implement a neural network in character recognition. Recently, with the development of parallel computation and the use of GPU, efficient OCR system based on neural network become realistic.

## 1.4 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 et al [9] 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 varied 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.

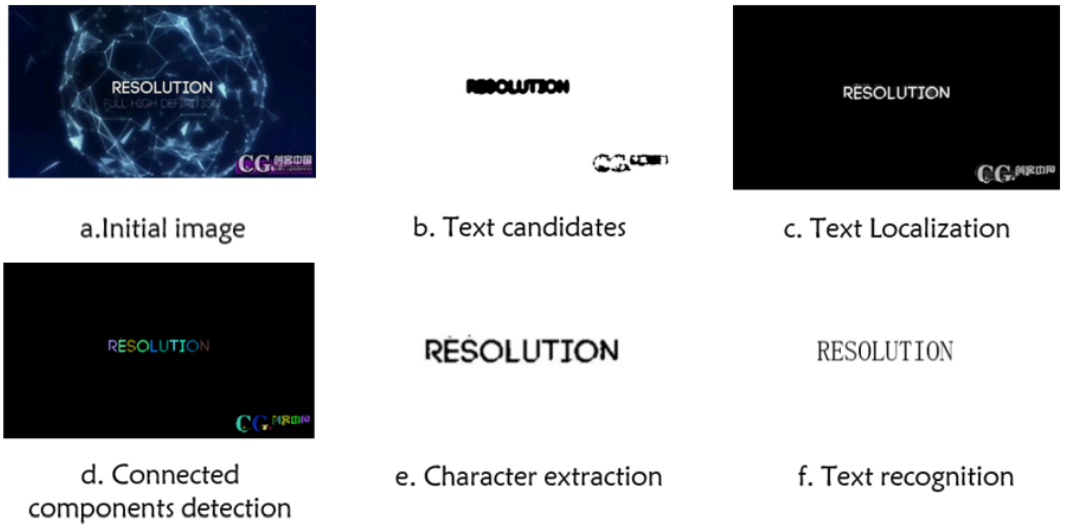



**Figure 2:** Overview

# 2 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 text recognition.

## 2.1 Contour extraction

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.

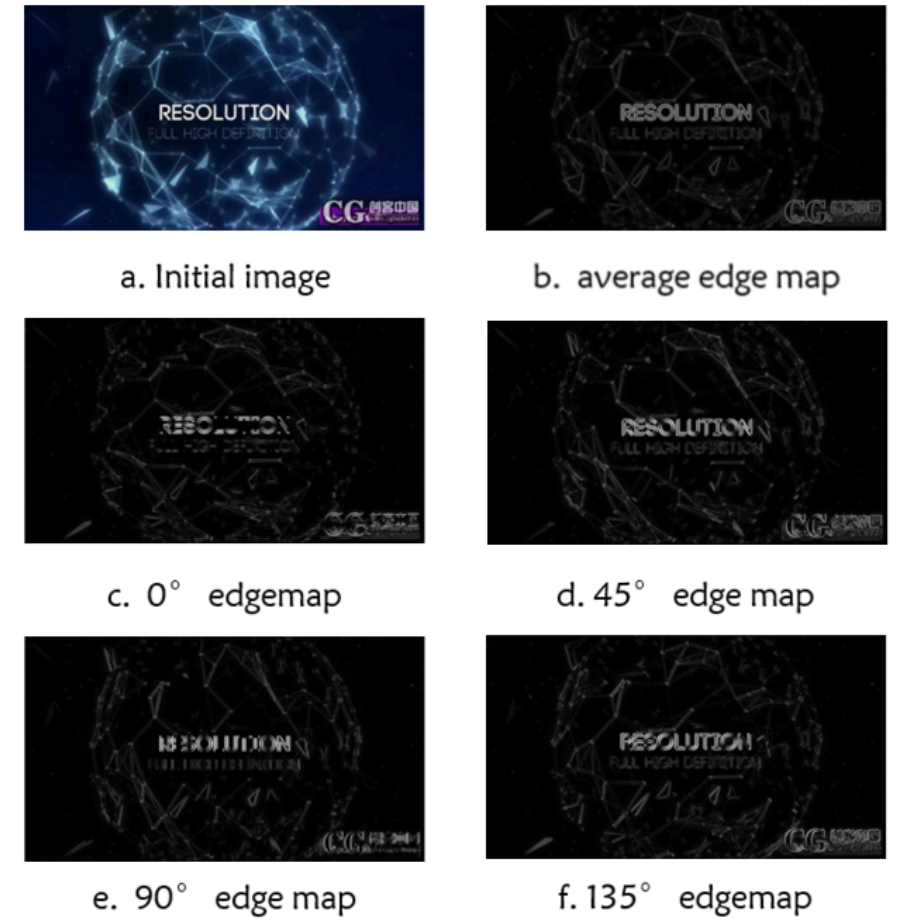



**Figure 3:** Edge maps

## 2.2 K-means clustering

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:

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

Where $E(i,j)$ is the value of pixel in i$^{\text{th}}$ row, j$^{\text{th}}$ column, here it is the grayscale color of a pixel in an edge map. $\mu$ is the mean value in sliding window, $\sigma$ is their standard deviation, $Eg$ is their energy, $Et$ 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 of 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.

## 2.3 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.

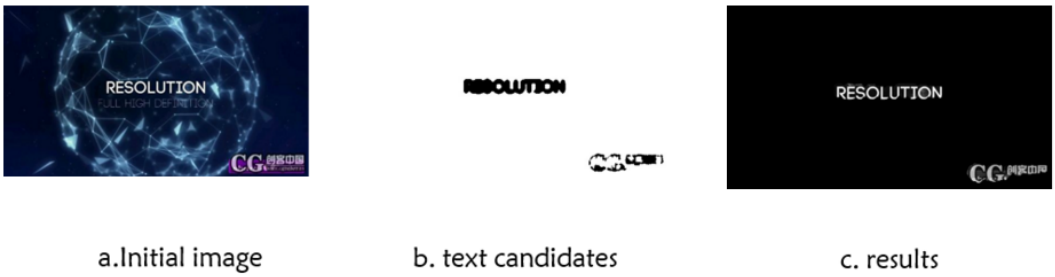



**Figure 4:** Text area identification

## 2.4 Connected components detection

The former method shows efficient results. However, it is still incapable of distinguishing 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 setting 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.

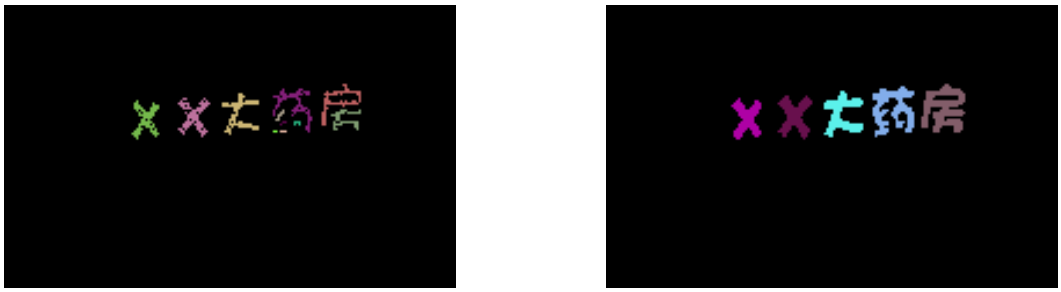

**Figure 5:** Connected components detection

### 2.5 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 $j$:
$\sqrt{||Gi - Gj||^2 + (greyscale_i - greyscale_j)^2}$

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.



**(a)** Simple connected components detection **(b)** Centered connected components detection

**Figure 6:** Components centering example

## 3 Text recognition

### 3.1 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 a 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 $\Theta$ by using gradient descent. At first, we need to initialize $\Theta$ by some random values in order to break the symmetry. Then, the forward propagation

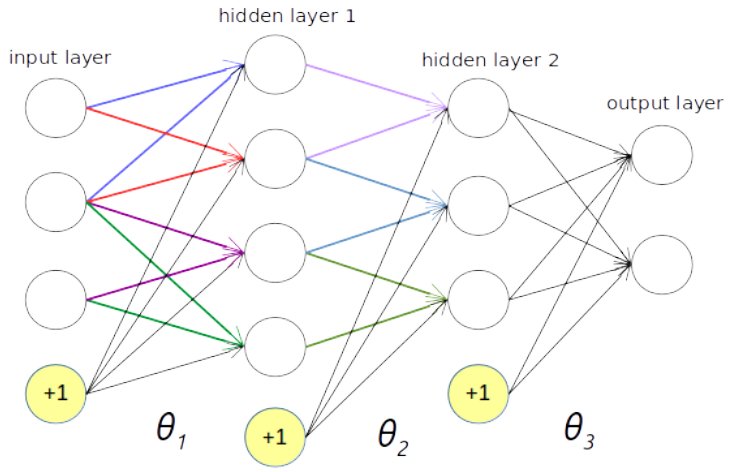



**Figure 7:** Structure of neural network

will determine the output while the backpropagation will correct the error.

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

### 3.2 Digit and letter recognition

In essence, pattern recognition convers the following problem: 'Given examples of signals and the correct decisions of them, make decisions automatically with future streams'. In our project, we use the neural network to learn a set of examples of digital numbers' images. Then we use the same neural network to predict new digital numbers' images.

For digital number recognition, we use MNIST dataset [8] 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 [4] where each letter 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.

## 4 Result

### 4.1 Text detection

Here are the text detection result with different photos:


**Figure 8:** Results on 3 images (from left to right: initial image, text area, character extraction)

## 4.2 Text recognition

We show you in this subsection some results and analysis of digit recognition. The MNIST dataset consists of 70.000 handwritten digit images of 0 to 9. We have tested the network with 10.000 different images and with different parameters of the network.

### 4.2.1 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..

### 4.2.2 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:

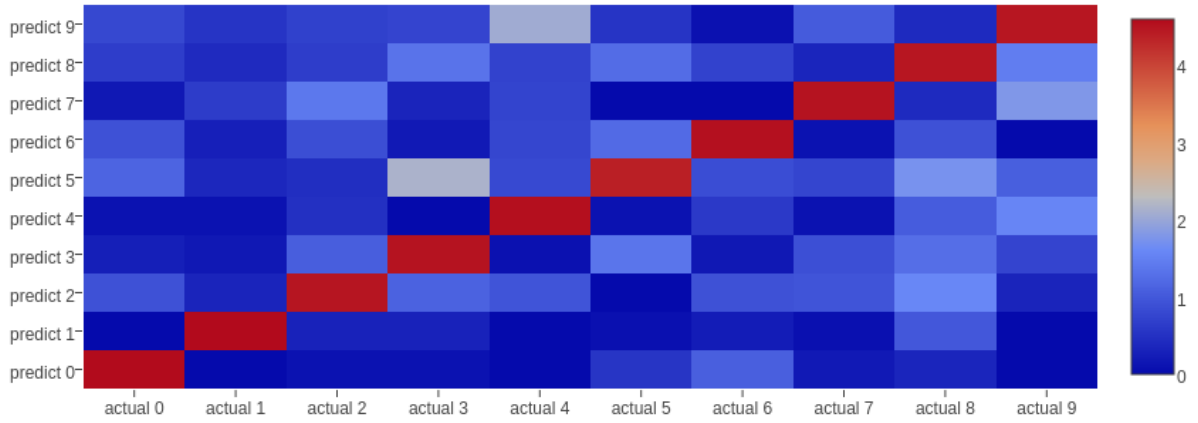



**Figure 9:** Heatmap of confusion matrix

Each color corresponds with $log\frac{number\_of\_predicted\_i}{number\_actual\_i}$, with this map, we can easily determine the error caused by the confusion of numbers. For exemple, the wrong prediction of number 3 is always 5 and the wrong prediction 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 in our system.

## 4.3 Remaining difficulties and possible improvements

There are several possible improvements to our projet:

1. As mentioned in section 3.2, after K-means clustering, we still need to determine which cluster represents text zone and which is the text. In our project, we adopte a manual selection approach. However, if we have an image dataset large enough, we can probably develop an automatic detection of these two clusters by training all images.

2. The connected component approach can glue up multiple letters, especially when letters are in hand-written style or with a serif font. What's more, hieroglyphic symbols like Chinese characters can sometimes break into pieces. Our centering approach can solve part of these problem but can also lead to wrong conclusion. An alternative method is to apply sliding window with well defined sizes and positions in different text zones to extract characters, or in case of words written in a line, directly cut it into segments and perform extration piece by piece.

3. The approach presented in this report only works with words without to much distortion. For example, letters in graffitis which are overlapped and painted with different pattern can not be located and extracted by our system. One possible way is to using sliding window to extract every part of the image and apply text recognition to all extractions.

# 5 Conclusion

We implement in this project a whole pipeline for content-based image understanding problem. Multiple machine learning techniques (K-means, ANN, etc.) and image processing techniques (Sobel edge detector, Ostu method, opening/closing, connected-components, etc.) have been combined in this approach for solving text detection and recognition problem.

Text detection is an important subject for machine-human interaction and artificial intelligence with great utility. We can imagine such a system used to help visually impaired people to read books and signs in the surroundings. It can also be used for automatic cars to understand road signs and react better in real time. Thanks to recent developments in electronics, computer science and data science, efficient implementations of this technique has been made possible.

# References

[1] Yuichiro Anzai. *Pattern Recognition & Machine Learning*. Elsevier, 2012.

[2] David J Burr. Experiments on neural net recognition of spoken and written text. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 36(7):1162–1168, 1988.

[3] Huizhong Chen, Sam S Tsai, Georg Schroth, David M Chen, Radek Grzeszczuk, and Bernd Girod. Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 2609–2612. IEEE, 2011.

[4] Teófilo Emídio de Campos, Bodla Rakesh Babu, and Manik Varma. Character recognition in natural images. In *VISAPP (2)*, pages 273–280, 2009.

[5] Boris Epshtein, Eyal Ofek, and Yonatan Wexler. Detecting text in natural scenes with stroke width transform. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2963–2970. IEEE, 2010.

[6] Nobuo Ezaki, Marius Bulacu, and Lambert Schomaker. Text detection from natural scene images: towards a system for visually impaired persons. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 683–686. IEEE, 2004.

[7] Kwang In Kim, Keechul Jung, and Jin Hyung Kim. Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(12):1631–1639, 2003.

[8] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[9] Chunmei Liu, Chunheng Wang, and Ruwei Dai. Text detection in images based on unsupervised classification of edge-based features. In *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, pages 610–614. IEEE, 2005.

[10] Simon M Lucas. Icdar 2005 text locating competition results. In *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, pages 80–84. IEEE, 2005.

[11] A Rajavelu, Mohamad T Musavi, and Mukul Vassant Shirvaikar. A neural network approach to character recognition. *Neural Networks*, 2(5):387–393, 1989.

[12] Qiviang Ye, Wen Gao, Weiqiang Wang, and Wei Zeng. A robust text detection algorithm in images and video frames. In *Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on*, volume 2, pages 802–806. IEEE, 2003.

[13] Xu-Cheng Yin, Xuwang Yin, Kaizhu Huang, and Hong-Wei Hao. Robust text detection in natural scene images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(5):970–983, 2014.

[14] Yu Zhong, Hongjiang Zhang, and Anil K Jain. Automatic caption localization in compressed video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(4):385–392, 2000.