International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 5 | ISSN : 2456-3307

Text Detection in Nature Scene Images

Arpitha G[1], Chaithra N .S[2], Greeshma B .R[3], Likitha Apoorva Varshini[4], Puneeth P[5]
[1-4]Information Science and Engineering, Maharaja Institute of Technology Mysore, Mysore, Karnataka, India
[5]Assistant Professor, Information Science and Engineering, Maharaja Institute of Technology Mysore, Mysore, Karnataka, India

ABSTRACT

Text detection in natural scenes holds great importance in the field of research and still remains a challenge because of size, various fonts, line orientation, different illumination conditions, weak character and complex background in image. Here a nature Text detection from nature images can be processed through the Raspberry Pi device. Raspberry Pi device is one of the controller and it acts like CPU and the image processing code ca be done to this device which runs the code. In our proposed method, input image is captured through 2 ways: one through the Pi cam (real time image) and other through the system(dataset). The images maybe of different font and colour, different slant degree and are of different format like jpeg, jpg, png, etc. All kind of images are accepted as input. Both the images are trained and supervised and hence training and testing images are same. These processes completely rely on Raspberry Pi. The given input images boundary is identified and is converted from RGB format to gray scale image for further processing. A gray scale image is obtained then text regions are obtained based on the morphological features. The clustering the characters is performed. Finally forming of words and output is processed data displayed on terminal. All the process run through the terminal and output is also displayed on terminal.

Keywords: Pi cam, Raspberry pi, RGB

I. INTRODUCTION

Text regions in images of natural scenes include important information for content-based image analysis such as web image search and video information retrieval. However, text regions in natural scene images usually vary greatly in font, size, colour, appearance and illumination, and can have complex backgrounds that differ from those in documents and business cards. These variations make text detection difficult in natural scene images. Automatic reading text from signs in natural images becomes a vital application in visually impaired people assistance. Its numerous potential applications can be described as image retrieval, scene understanding, visual assistance, etc. However, finding the text in scene images is a great challenge, because it cannot be assumed that the acquired image contains only characters. Natural scene images usually contain diverse complex text of different size, styles and colours with complex backgrounds.

Scene text detection aims to locate the position of text in different scenes, for example a guide post, store marks, and warning signs which is one of the most important steps for end to end scene text recognition. Effective scene text detection can enhanced the performance of numerous multimedia applications example mobile visual search, content-based image retrieval and sign automatically translation. Scene text can appear with any slant, tilt, in any light and upon any surface and hence hard to detect, localize and recognize. There are number of pre-processing steps followed by the actual recognition in the implementation of text recognition system. The paper follows. In section 2 it describes about the raspberry pi and method used, section 3 describes about the accuracy result of the proposed method, section 4 describes about the conclusion and feature work.

II. METHODS AND MATERIAL

To obtain text from the nature Scene image it goes with implementation of 4 algorithms.

Four basic algorithms:

The algorithm relies on a set of learned characters. It compares the characters in the scanned image file to the characters in this learned set

- Image labelling.
- Finding boundary and Generating X, Y coordinate pixel array.
- Matching connected pixels with learned set (in the given image).
- Forming words.

Image labelling algorithm:

It uses the Two-pass algorithm, which relatively simple to implement and understand, the two-pass algorithm iterates through 2-dimensional, binary data. The algorithm makes two passes over the image: one pass to record equivalences and assign temporary labels and the second to replace each temporary label by the label of its equivalence class.

Finding boundary and Generating X, Y coordinate pixel array.

From the labels from the above algorithm, then its merely adding all the connected X, Y coordinates in the connect component list. The above image shows all the connected component boundary which marked in yellow. I have highlighted the boundary (X, Y) coordinates of the connected component "a".

LeftXCor: - Starting left X coordinate of the connected component. For the connected component "a" it is 0.

RightXCor: - Ending left X coordinate of the connected component. For the connected component "a" it is 4.

TopYIndex: - Starting or the lowest Y coordinate of the connected component. For the connected component "a" it is 0.

BottomYIndex: - Ending or the highest Y coordinate of the connected component. For the connected component "a" it is 5.

Width: - Width of the connect component will be RightXCor – LeftXCor.In case of "a" it will be 4 - 0 = 4 But add one since it start from zero so the width will be 5.

Height:-Similarly height of the connected component will be BottonYCor – TopYCor.In this case for "a" the height will be 6.

bitmap image and pixel information in the image for character "a". As we see the boundary in yellow line. In above diagram the first pixel coordinate (0,0) where X and Y coordinates are zero. As explained earlier the boundary conditions. The properties and their values are listed below.

LeftXCor: - Starting left X coordinate of the connected component. For the connected component "a" it is 0.

RightXCor: - Ending left X coordinate of the connected component. For the connected component "a" it is 4.

TopYIndex: - Starting or the lowest Y coordinate of the connected component. For the connected component "a" it is 0.

BottomYIndex: - Ending or the highest Y coordinate of the connected component. For the connected component "a" it is 5.

Width: - Width of the connect component will be RightXCor – LeftXCor.In case of "a" it will be 4 - 0 = 4.But add one since it start from zero so the width will be 5.

Height:-Similarly height of the connected component will be BottonYCor – TopYCor.In this case for "a" the height will be 6.

Note the connected X, Y coordinate in the xml.For "a" (0,3)(0,4) etc pixels are high so they are noted down.X,Y coordinates whose pixels are not high they are not noted.The tag <pixelinfo> represent the pixel coordinates whose pixels are high. The <ParamValue> tag has the character value "a".

Matching character:

Finally this most easy task. We match the connect component bit array with the xml data. Each pixel are matched according the X, Y coordinates. The fully matched pixels coordinates is the matched character from the xml.

Forming words:

As per the above example "Basic". We maintain the LeftXindex and RightXindex for each character. The LeftXindex represent the left most index of the character in the bitmap specified initially in the blog. The RightXindex represent the right most X coordinate of the character. When the difference coordinates of current character and previous character is less than 3 pixels then they are joined. This algorithm is quite simple. But you can extend to join words according to grammar in the dictionary.

Software Requirements

- Operating System: NOOBS
- Programmed language: Python3

Hardware Requirements

- Raspberry Pi
- Pi Cam
- Display
- Input Devices [key board, Mouse]

NOOBS:

NOOBS is designed to make it easy to select and install operating systems for the Raspberry Pi without having to worry about manually imaging SD card.

Python

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python is Interpreted: Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

Python is Interactive: You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python is Object-Oriented: Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

Python Features

- **Easy-to-learn:** Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read:** Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain:** Python's source code is fairly easy-to-maintain.
- **A broad standard library:** Python's bulk of the library is very portable and crossplat form compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode**: Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **GUI Programming:** Python supports GUI applications that can be created and ported to many system calls, libraries, and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

Raspberry Pi

Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation. Several generations of Raspberry Pis have been released. All models feature a Broadcom system on a chip (SoC) with an integrated ARM compatible central processing unit (CPU) and on-chip graphics processing unit (GPU).

Processor speed ranges from 700 MHz to 1.4 GHz for the Pi 3 Model B+; on-board memory ranges from 256 MB to 1 GB RAM. Secure Digital (SD) cards are used to store the operating system and program memory in either SDHC or MicroSDHC sizes. The boards have one to four USB ports. For video output, HDMI and composite video are supported, with a standard 3.5 mm phono jack for audio output. Lower-level output is provided by a number of GPIO pins which support common protocols like I²C.

Pi Cam:

PiCAM is an autonomous high definition video system with an embedded web server. It uses the Omnivision 5647 sensor which offers a detailed resolution up to 5MP (megapixel). The PiCAM provides multi-day recordings, depending on the memory capacity of the SD card used for on-system storage.

III. RESULTS AND DISCUSSION

Sr. no	Parameters	Proposed Application
1	Applicable areas.	In computer Applications
2	Operating systems on which it is accessible.	Raspberry pi OS: NOOBS
3	Platform status	Platform dependent
4	Back-End	Python
5	Front-end	Raspberry Pi
6	Middle-End	Command prompt
7	Technology used	Raspberry Pi Communicates with the hard wares.
8	Efficiency	Higher accuracy.
9	Advantage	Minimal or null changes are required for implementing this system.
10	Disadvantage	It is not platform independent which makes it accessible only to particular systems.

RESULT DISCUSSION:

The Application detects and extracts the text content from natural scene images and displays the exact text on console, where raspberry pi module processes it. The future work that one can carry out as an extension of this model is to used for detecting the text from some complex background images. Also by increasing the image enhancement we can extract text in an more accurate manner.

IV. CONCLUSION

Text detection from nature Scene images has lot more challenges like illumination, colour, font size, slant angle etc. Our proposed Method Identifies the text with different background, different colour, different font style along with little skewed text. Since proposed method Uses the Raspberry pi as the controller all the processes are quick and reliable. Scene images with text are pre-processed hence the results are of better Accuracy. Major drawback of the proposed Method is failed detect and extract the text with smaller font size.

V. REFERENCES

[1]. Jianjun Chen Department of Intelligent Systems Design Engineering, IEEE,2017. A Homogeneous Region Based Methodology for Text Extraction from Natural Scene Images.

[2]. Myung-Chul Sungl, Bongjin Jun2, Hojin Ch02 and Daijin Kiml IEEE,2015. Scene Text Detection with Robust Character Candidate Extraction Method