Title: Text Extraction and Identification From Images

Course: UE14CS348 - Digital Image Processing

Authors: Shruthi Ramesh(1), Shwetha Srinath(2)
SRN: (1) 01FB14ECS236 ; (2) 01FB14ECS239
Section D, P E S University
Bangalore,India
(1)shruthiramesh8@gmail.com ; (2)shwetha1729@gmail.com

***Abstract*—This document is a report on our project for the course Digital Image Processing-UE14CS348 built during the course of Jan-Apr 2017. The project is an attempt to extract and recognise the text present in a given image. We use the technique of template matching to arrive at a reasonable conclusion of whether the image contains text in a particular language.**

***Keywords*—optical character recognition, text extraction, template matching**

I. Introduction (*Heading 1*)

The problem of recognition of text from images is a classic application of image processing.
It becomes more and more necessary to develop high-accuracy techniques to extract text from images and obtain useful information from the data thus extracted.
With the tools we have at hand, we attempted to extract this information using the techniques of finding contours and template matching. We used the library openCV[1] with Python to do the same.

II. Techniques used

A. *Thresholding*

The simplest thresholding methods replace each pixel in an image with a black pixel if the image intensity is less than some fixed constant T, or a white pixel if the image intensity is greater than that constant. In the image below, this results in the tree becoming completely black, and the white snow becoming completely white. [2]

In our project, we use binary thresholding for the simple reason that we require text from the image which is more or less at a fixed range of intensities as compared to the background intensity range. Hence, binary thresholding is sufficient and produces satisfactory results.

B. *Contouring*

A contour is defined as a curve joining all the continuous points (along the boundary), having same color or intensity.

The purpose of finding contours in our implementation is to isolate the different parts of our image and separate out the letters from the background. An example from the outputs we obtained is shown below. The bright green boundaries are the contours found by the findContours function in openCV.

C. *Convolution*

We use the concept of convolution in the matchTemplate function to compute the similarity between each of the segments obtained from contouring and our pre-defined template images. Convolution is a technique which assists us in making a rudimentary analysis of how close the input image segment is to the template image.

Mathematically defined, convolution is the process of flipping both the rows and columns of the kernel and then multiplying locationally similar entries and summing.[3]

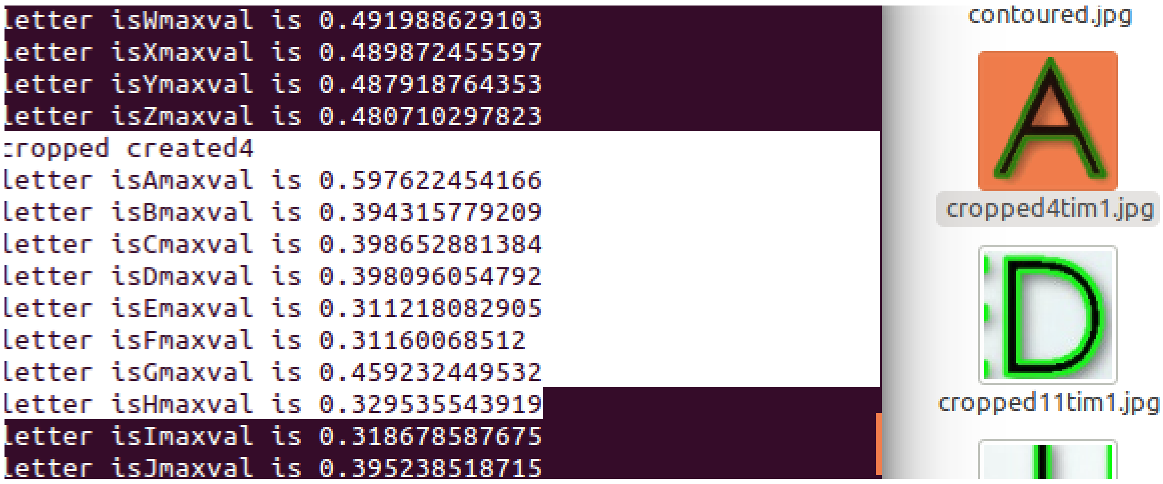
D. *Template Matching*

Here, we use template matching using the process of convolution to determine whether the segment of the image is likely to be the letter that we are trying to find using a template. Once we obtain the result of the convolution, we use the minmaxLoc function to arrive at a localized point that has the highest correlation with our template. Now that we have found this point, we draw a rectangle around it of desired width(obtained by empirical analysis , image dependent) and recognise that as the matched portion of the image.

Outputs and Observations

A. *Results*

We were able to arrive at fairly satisfactory results with a very good probability of some letters occurring and some other slightly disappointing likelihoods. The analysis of our results helped us make some observations which are detailed in the next subsection. In brief, we concluded that template matching [4] would work, however only for a very limited number of samples and we require a better technique to be able to handle this issue of scalability. An illustration of the probabilities observed when we ran our code can be found below.



```
letter isWmaxval is 0.491988629103
letter isXmaxval is 0.489872455597
letter isYmaxval is 0.487918764353
letter isZmaxval is 0.480710297823
cropped created4
letter isAmaxval is 0.597622454166
letter isBmaxval is 0.394315779209
letter isCmaxval is 0.398652881384
letter isDmaxval is 0.398096054792
letter isEmaxval is 0.311218082905
letter isFmaxval is 0.31160068512
letter isGmaxval is 0.459232449532
letter isHmaxval is 0.329535543919
letter isImaxval is 0.318678587675
letter isJmaxval is 0.395238518715
```

B. Observations

The technique of template matching is highly rudimentary and works perfectly on extremely few cases. We realised the necessity of using some sort of machine learning technique to implement our OCR for language recognition, which is the first step of any future improvements we may make to the project. However, it was enticing to observe probabilities that were very well in accord with what we expected for our limited test data against the templates. Our research led us to realise that neural networks were more suited to this application and produced far better results, and we would like to delve deeper into the same in order to obtain a better identifier for text in images.

Acknowledgment

We would like to sincerely thank our professor, Dr. Gowri Srinivasa for the wonderful explanations of the various techniques used in image processing, which really helped us during our project and seminar.

In addition, we express our heartfelt thanks to our classmates for the seminars that were extremely informative and helped us throughout our endeavours in this project. We thoroughly enjoyed the course of UE14CS348 and hope to come up with better solutions to problems in the domain of image processing in future, using our current knowledge.

References

[1] http://docs.opencv.org/2.4/doc/tutorials/tutorials.html
[2] https://en.wikipedia.org/wiki/Thresholding_(image_processing)

[3] http://mathworld.wolfram.com/Convolution.html

[4] http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html