iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** V **Month of publication:** May 2017

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

Volume 5 Issue V, May 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

An Introduction to Optical Character Recognition

Shashank[1], Nishant Parkey[2], Shruti Keshari[3]
Students, Bachelors of Technology Deptt. of Computer Science & Engineering IMSEC, Ghaziabad, U.P., India

Abstract: This article is all about the development of OCR usingMATLAB. It describes all from segmentation toclassification. To understand how an experimental approach was chosen, for example, the result got affected by the use of different properties of the characters.
Key words: OCR, feature extraction, character classification, neural network

I. INTRODUCTION

Optical Character Recognition make brief to OCR.
Let us imagine that you have a pile of printed papers, the task is to convert that text into digital format. What to do now? Either type every single word into a word processor or make it simpler by using an OCR program. Typing the whole thing would be a tiresome and big time consuming task whereas, with an OCR program you can feed with your scanned document. The OCR program processes the scanned image and then detect the letters. The letters are then sent to the word processor and the job is done!
As mentioned earlier the goal of this article is to conclude that how such an, simple but working, OCR algorithm can be implemented. To head towards the goals and in order to match the given task some limitations are to be made to the abilities of the program.
A short list of such limitations is as follows:

A. The letters from a to z are recognized only.
B. If the text is having several lines the left most letters of every line will be treated as the first character by the program i.e, even if the original text consists of several lines the program will return a single line of text.
C. Our focus is on the typeface Times New Roman.
D. As in practice it is not possible to process too small characters, the letters must not be connected in the image.

This technique of recognizing the characters from a digitally scanned image projects many aspects of image analysis and processing. The approach we have worked on in this matter is explained in the methods section followed.
There has been no thought of performance in terms of speed in the process. Instead our work was based on the analysis and the processing part. As the main aim was not to make a commercial program but instead a very simple working character recognition.

II. METHODS

To have a glance on what sort of methods that we are using, let us have a look at what we do in our code.
The OCR program consists of the following processes:

A. Segmentation
B. Feature extraction
C. Classification

By using a threshold, we convert our original scanned image into a binary image, only having zeroes and ones.
Next we look at each pel and its neighbours. The way of looking at the pixels neighbours is 8-connected. We give the segment a number, when we have found all the pels that got connected to each other.
A very interesting and sound function of MATLAB is *imfeature*. This function enables us to extract a lot of data from each segment. We can look efficiently at each segment in different ways, from the data calculated bt*imfeature* function. The very first way is to look at each segment's Euler number. A segment is assigned the value 1. 1 is subtracted, for every hole in the segment. As an example 'a'. It has one hole in it, so the Euler number for 'a' is 0. For 'g' , the Euler number is -1.
The second method we can look at each segment, is by dividing every segment into four equally sized rectangles. Then the relative area for each section is looked. The term relative area can be defined as the sum of pels occupied in the segment divided by the whole area of that rectangle. This is a way to find out if the searched letter is 'b' or 'd' and 'q' or 'p'.

Also we look at the form of the segment. We do this by dividing the height by the width of the segment. This is the way to differentiate between 'h' or'n'.
The feature vector then have the six elements which are all invariant to the resolution of the segments as under:

Euler number
Height divided by width
Upper lest intensity
Upper right intensity
Lower lest intensity
Lower right intensity

Now we some character characteristics. The neural network data is feed with this data. The likeliness of all the character is evaluated by the net known as system. It selects the one with the highest value in the system.
Early as said we don't take care of character other than a to z in this version. The output excludes the dots, but the program recognizes them. In the upcoming version, there would be an improvised method to check if the dots is a period by looking in the vertical position.
A simplest version of white space detection is being used in the program. mean width of the characters is being calculated. Then if the length from the right side to the next letters left side is greater than half of the mean value, we have a white spacing between the two letters.
NEURAL NETWORKS: A feedback network consisting of 3 layers is used and the size is as per the experiments. The network was trained for about 600 000 epochs consisting of 300 neurons with 6 inputs and 27 outputs.
Till now, the MATLAB has the finest way of creating a neural network. The function is called newff. A new image is made with letters from 'a' to 'z', and known as the inputs for the network. The matching letters are also defined as the output. The network is then trained and made to distinguish between the letters. To know more see help *neuff* in MATLAB.

III. EXPERIMENTS

We wanted to add some functionality and different abilities in order to improve the programs when the basic functions are implemented. This were done by experimenting with,for example the character characteristics and the structure of the neural network.
Use of many characters were nit there as correct way to it was only checked. The program succeeded in finding the character, even though they were not correct.
Certain characteristics were removed and added as the development proceeded. For example, a couple of moment equations were added to the recognition. While the training of the neural network it turned out to give poor results. It can be due to the poor implemented moment calculation the network wasn't right.
The characteristics were made independent of size of the scanned text. It would be great while scanning the text of different sizes and resolution. The network with the whole alphabet in 5 various sizes were trained for 3 days and saw that it suffered badly because of the tuned network. No solutions were made after so much of potentials. While on the other hand, the program recognizes text in diverse sizes properly.
The biggest flaws were added to the white spacing so that it can read text with lines in correct order. The letters are processed from left to right without any consideration to which it fall.
We chose to include the characteristics described in section methods above with weeks of experimenting behind us. A nice estimation of scanned characters for this assignment is provided by them at affordable speed.

IV. CONCLUSION

It is shown by this article that a well-functioning OCR system can be developed by using very minimal codes, with a limited domain knowledge and within a short period of time. This can be made possible due to powerful support functions provided by the toolbox in MATLAB and fast computers which increases the learning speed of the system.
With the provided program the system is able to solve problems containing one line. To be used practically the minimal requirement of a OCR program would be recognising a full page of text in a very few seconds. The program we have designed will probably fail that test because it is not optimized for speed.

An exponential relation can be observed between the size of the problem domain and the complexity required by it. We can perhaps dictionaries and other higher level information in order to maximize accuracy, to develop a commercial system. Since it is not that easy to recognize texts with variations in size and fonts, our program is designed to detect and recognize text with given size and font.

V. FUTURE WORK

If you ask for a simple OCR system our program will provide you the basic things. This is the fact that we would not be able to sell our program to people for a lot of money. For commercial purposes we have to implement some extra features to the program. Some of those extra additions to our program are as follows:

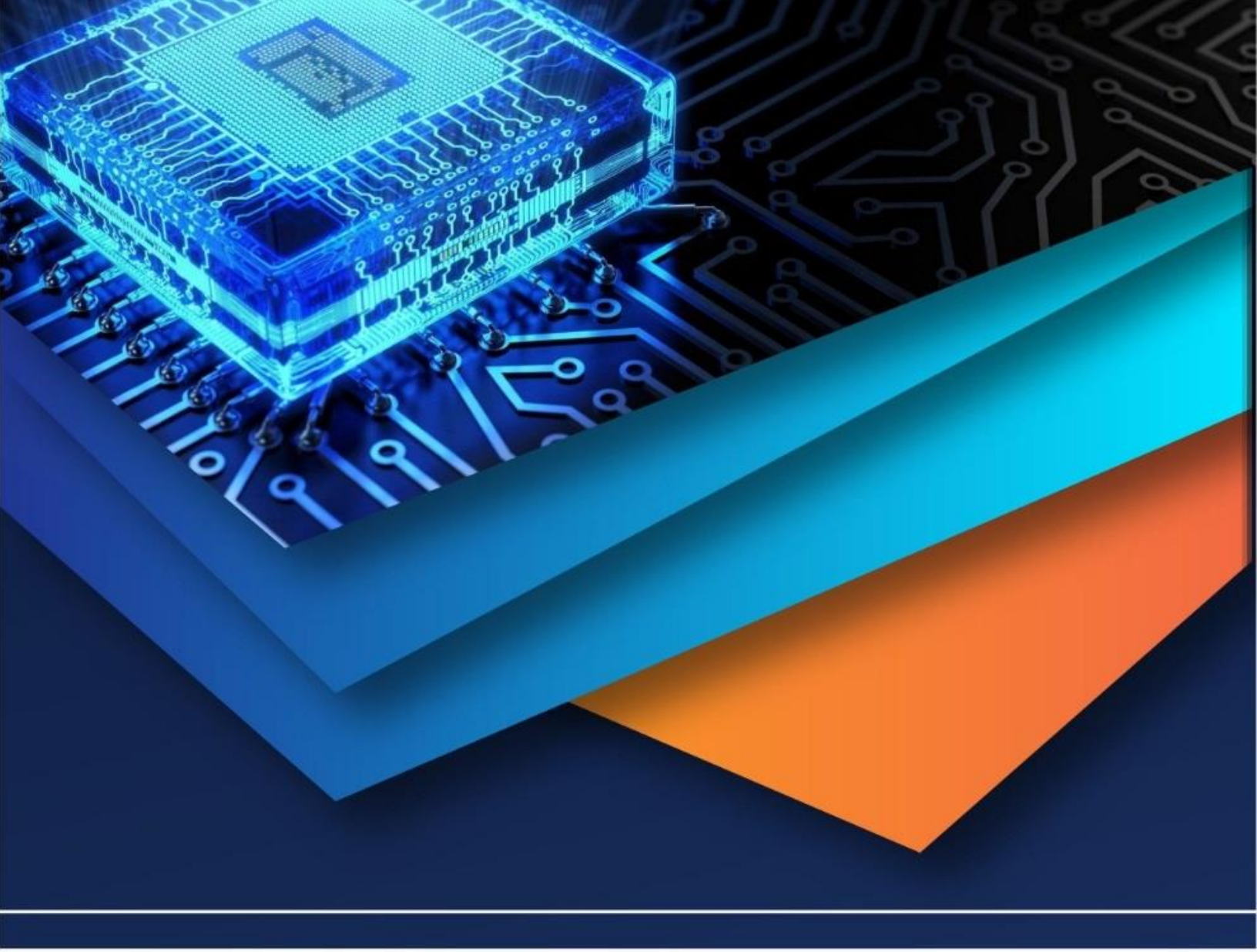
A. A feature that could handle period characters i.e; would be able to detect other letters than a to z.
B. A feature which can scan several rows of lines simultaneously.

VI. ACKNOWLEDGEMENT

We thank everybody involved for an interesting course in image analysis and OCR.

REFERENCES

[1] https://www.researchgate.net/publication/267156858_A_Survey_of_Telugu_OCR_System
[2] http://www.academia.edu/5209270/An_Overview_of_OCR_Research_in_Indian_Scripts
[3] Help sections in MATLAB v.6



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY