TEXT RECOGNITION (OCR) USING DIGITAL IMAGE PROCESSING TECHNIQUES

PROJECT REPORT

Submitted in fulfilment for the JComponent of

Digital Image Processing (SWE1010)

CAL Course

In

M.Tech. – Software Engineering

By

K.KAVYA- 16MIS0084

Under the guidance of

Dr. Hemalatha S

SITE

School of Information Technology and Engineering

Winter Semester 2018-19

TABLE OF CONTENTS

ABSTRACT:

The objective of text recognition is to recognize the text from printed hardcopy document to preferred format. recognizing a text is a very easiest job for humans, but in a machine or electric device that does text recognition is a challenging task. the major steps involved in text recognition is pre-processing, segmentation, feature extraction, classification. to attain high speed in data processing it is essential to convert the analog data into digital data. The process of Text Recognition involves several steps including preprocessing, segmentation, feature extraction, classification, post processing. Preprocessing is for done the basic operation on input image like binarization which convert gray Scale image into Binary Image, noise reduction which remove the noisy nature from image. Segmentation stage for segment the given image into line by line and segment each character from segmented line. Future extraction calculates the characteristics of character. A classification contains the database and does the comparison. storage of hard copy of any document occupies large space and retrieving of information from that document is time consuming.

INTRODUCTION:

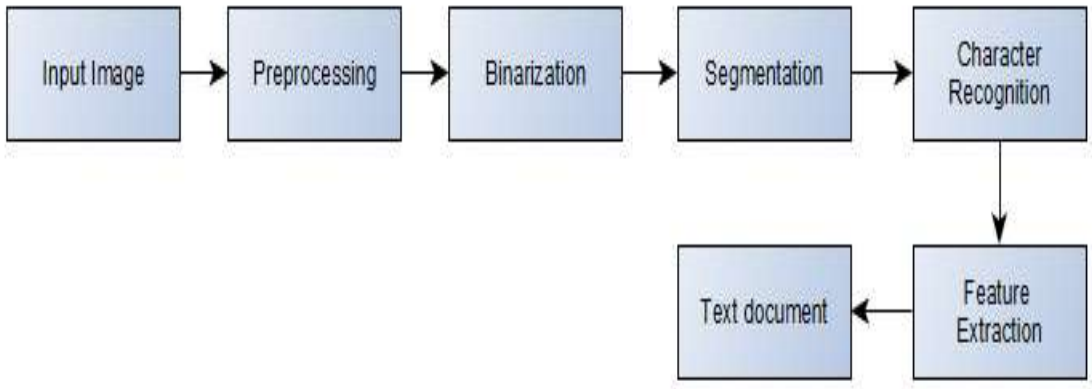
Nowadays all over digitization technology is used. Text Recognition usually abbreviated to OCR, involves a computer system designed to translate images of typewritten text (usually captured by a scanner) into machine editable text or to translate pictures of characters into a standard encoding scheme representing them. OCR began as a field of research in artificial intelligence and computational vision . Text Recognition used in official task in which the large data have to type like post offices, banks, colleges etc., in real life applications where we want to collect some information from text written image. People wish to scan in a document and have the text of that document available in a .txt or .docx format. in this project will recognize text that are in images using an open source tool called **tesseract** and opencv. the method of extracting text from images is also called optical character recognition (**ocr**)

or sometimes simply text recognition. Optical character recognition system is an effective way in recognition of text. OCR affords the easiest way to recognize and convert the printed text on image into the editable text. It also increases the speed of data reclamation from the image.There are numerous applications in which text extraction is useful. Latest technology in the field of image processing express a great amount of interest in content retrieval from images and videos. Character recognition is an art of detecting segmenting and identifying characters from image. More precisely Character recognition is process of detecting and recognizing characters from input image and converts it into ASCII or other equivalent machine editable form [1], [2], [3].

PROBLEM STATEMENT:

OCR technology has been used to convert the text in scanned paper documents into ASCII symbols .However, current commercial OCR systems do not work well if text is printed against shaded or hatched backgrounds, often found in documents such as photographs, maps, monetary documents ,engineering drawings and commercial advertisements .Furthermore ,these documents are usually scanned in greyscale or color to preserve details of the graphics and pictures which often exist along with the text. For current OCR systems, these scanned images need to be binarized before actual character segmentation and recognition can be done. A typical OCR system does the Binarization to separate text from the backgrounds by global thresholding .Unfortunately, global thresholding is usually not possible for complicated images, as noted by many researches. Consequently, current OCR systems work poorly in these cases. One solution to the global thresholding problem is to use different thresholds for different local regions(adaptive thresholding)

SYSTEM DESIGN



3.SOFTWARE REQUIREMENT SPECIFICATIONS

PYTHON –ANACONDA NAVIGATOR

SPYDER:

Spyder is a powerful scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. It features a unique combination of the advanced editing, analysis, debugging and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection and beautiful visualization capabilities of a scientific package.n Furthermore, Spyder offers built-in integration with many popular scientific packages, including NumPy, SciPy, Pandas, IPython, QtConsole, Matplotlib, SymPy, and more.n Beyond its many built-in features, Spyder can be extended even further via third-party plugins.n Spyder can also be used as a PyQt5 extension library, allowing you to build upon its functionality and embed its components, such as the interactive console or advanced editor, in your own software.

4.IMPLEMENTATION DETAILS

4.1 SOURCE CODE

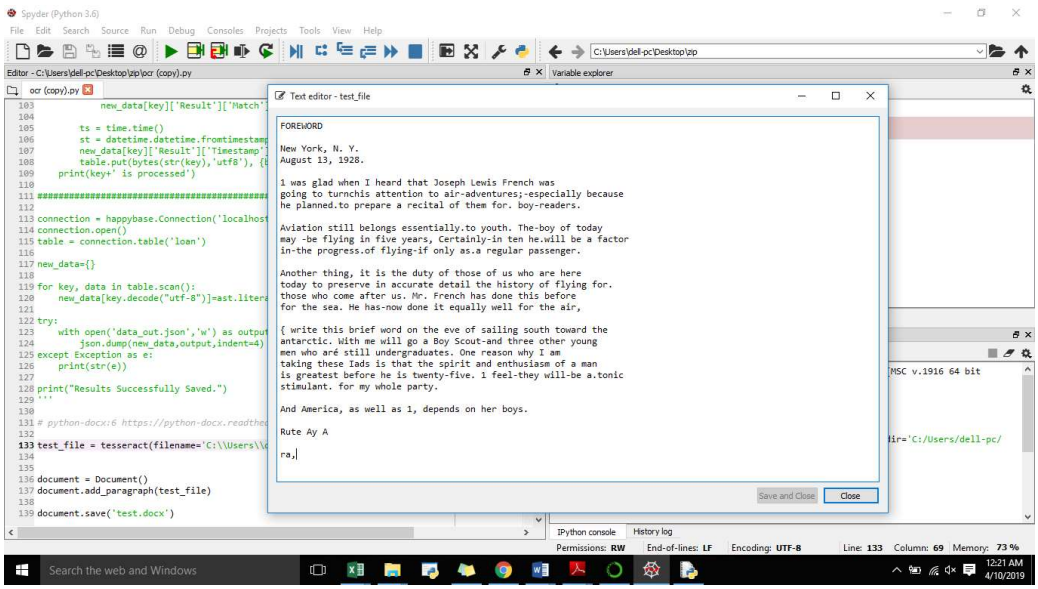
```
import pytesseract
import cv2
import os
import numpy as np
from docx import Document
from PIL import Image,ImageEnhance
def tesseract(filename):
im = Image.open(filename)
rgb_im = im.convert('RGB')
rgb_im.save("test.jpg", dpi=(300,300))
image_dpi = cv2.imread('test.jpg',0)
os.remove("test.jpg")


blur = cv2.bilateralFilter(image_dpi,15,75,75)
```

```
th3=cv2.adaptiveThreshold(blur,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,cv
2.THRESH_BINARY,11,2)

ret3,th3cv2.threshold
(th3,0,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C+cv2.THRESH_OTSU)

imagem = cv2.bitwise_not(th3)

  kernel = np.ones((1,1),np.uint8)

  eroded_img = cv2.erode(imagem,kernel,iterations = 1)

    #canny=cv2.Canny(imagem,100,200)

     config = ('-l eng --oem 1 --psm 3')

 text = pytesseract.image_to_string(eroded_img, config=config)

 return text

test_file = tesseract(filename='C:\\Users\\dell-pc\\Desktop\\scan4.png')

document = Document()

document.add_paragraph(test_file)

document.save('test.docx')
```

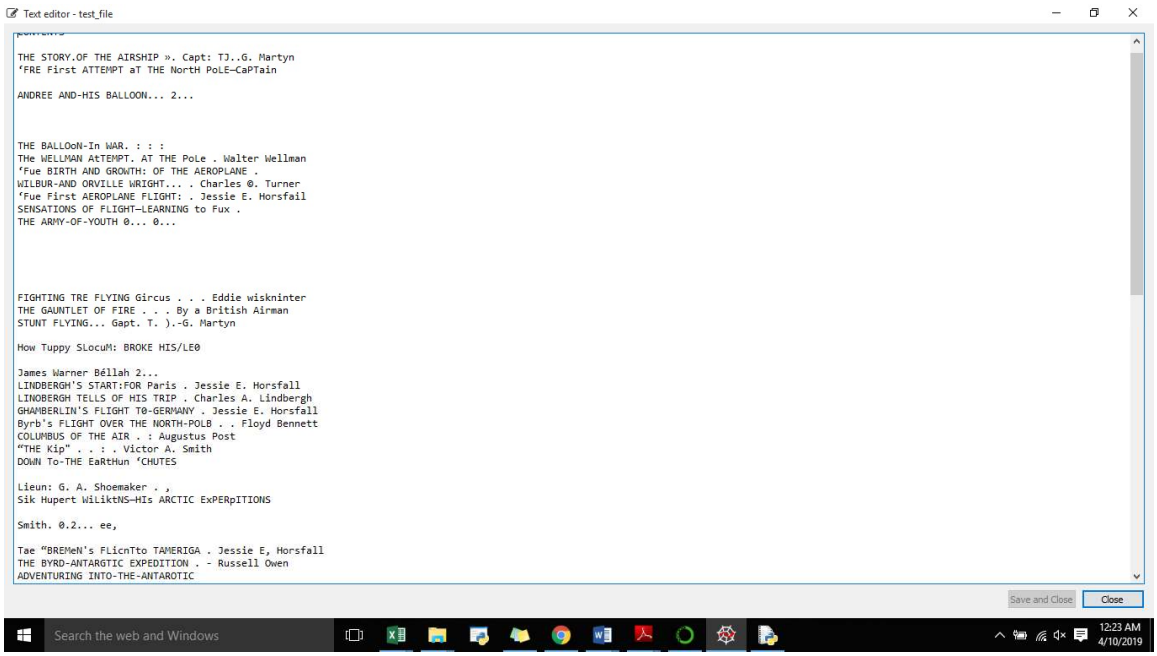
5.RESULTS

5.1 Screenshots

Scan1.jpg



Scan2.jpg



6.BASE REFERENCE PAPER

http://www.ijcea.com/wp-content/uploads/2018/02/NCDCM_2017_paper_55.pdf

7. REFERENCES

[1] Kai Ding, Zhibin Liu, Lianwen Jin, Xinghua Zhu, "A Comparative study of GABOR feature and gradient feature for handwritten 17hinese character recognition", International Conference on Wavelet Analysis and Pattern Recognition, pp. 1182-1186, Beijing, China, 2-4 Nov. 2007

[2] Pranob K Charles, V.Harish, M.Swathi, CH. Deepthi, "A Review on the Various Techniques used for Optical Character Recognition",International Journal of Engineering Research and Applications, Vol. 2, Issue 1, pp. 659-662, Jan-Feb 2012

[3] Om Prakash Sharma, M. K. Ghose, Krishna Bikram Shah, "An Improved Zone Based Hybrid Feature Extraction Model for Handwritten Alphabets Recognition Using Euler Number", International Journal of Soft Computing and Engineering, Vol.2, Issue 2, pp. 504-58, May 2012

[4]Rashid, S. F., Shafait, F., & Breuel, T. M. (2012, March). Scanning neural network for text line recognition. In *2012 10th IAPR International Workshop on Document Analysis Systems*(pp. 105-109). IEEE.[4]

[5]. Wick C, Reul C, Puppe F. Improving OCR Accuracy on Early Printed Books using Deep Convolutional Networks. arXiv preprint arXiv:1802.10033. 2018 Feb 27.

[6] Breuel, T. M. (2017, November). High performance text recognition using a hybrid convolutional-lstm implementation. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)* (Vol. 1, pp. 11-16). IEEE.

[7] Cheng, Z., Bai, F., Xu, Y., Zheng, G., Pu, S., & Zhou, S. (2017). Focusing attention: Towards accurate text recognition in natural images. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 5076-5084).

[8] Sankaran, N., & Jawahar, C. V. (2012, November). Recognition of printed Devanagari text using BLSTM Neural Network. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)* (pp. 322-325). IEEE.[8]

[9] Ray, A., Rajeswar, S., & Chaudhury, S. (2015, January). Text recognition using deep blstm networks. In *2015 eighth international conference on advances in pattern recognition (ICAPR)* (pp. 1-6). IEEE.[9]

[10]. Su, B., & Lu, S. (2017). Accurate recognition of words in scenes without character segmentation using recurrent neural network. *Pattern Recognition*, *63*, 397-405.