



**KALASALINGAM**  
**ACADEMY OF RESEARCH AND EDUCATION**  
**(DEEMED TO BE UNIVERSITY)**  
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A" Grade



MINI PROJECT

# Optical Character Recognition using Java

(Submitted in partial fulfillment of requirements for the Course CSE18R272 - JAVA PROGRAMMING)

Submitted by

**YAMSANI RAVI TEJA-9918004165**

**MADADI SAINATH REDDY-9918004162**

Under the guidance of

**Mr.A.BHUVANESHWARAN**



April 2020

# Contents

<b>1</b>	<b>ABSTRACT</b>	<b>1</b>
<b>2</b>	<b>CANDIDATE DECLARATION</b>	<b>2</b>
<b>3</b>	<b>ACKNOWLEDGEMENT</b>	<b>3</b>
<b>4</b>	<b>TABLE OF CONTENTS</b>	<b>4</b>
4.1	chapter 1 INTRODUCTION . . . . .	4
4.2	chapter 2 PROJECT DESCRIPTION . . . . .	5
	4.2.0.1 HISTORY OF TESSERACT . . . . .	5
	4.2.1 ARCHITECTURE OF TESSERACT . . . . .	5
4.3	WORKING . . . . .	6
4.4	CONCLUSIONS . . . . .	9
<b>5</b>	<b>REFERENCES</b>	<b>10</b>

# 1. ABSTRACT

Optical character recognition (OCR) method has been used in converting printed text into editable text. OCR is very useful and popular method in various applications. Accuracy of OCR can be dependent on text preprocessing and segmentation algorithms. Sometimes it is difficult to retrieve text from the image because of different size, style, orientation, complex background of image etc. We begin this paper with an introduction of Optical Character Recognition (OCR) method, History of Open Source OCR tool Tesseract, architecture of it and experiment result of OCR performed by Tesseract on different kinds images are discussed. We conclude this paper by comparative study of this tool with other commercial OCR tool Transym OCR by considering vehicle number plate as input. From vehicle number plate we tried to extract vehicle number by using Tesseract and Transym and compared these tools based on various parameters.

## 2. CANDIDATE DECLARATION

The work reported in this has not been submitted by me for the award of any other degree of this or any other institute. *I hereby declare that the work presented in this report entitled “Optical Character Recognition using Java”, in partial fulfilment of the requirements for the award of the degree Bachelors in Technology and submitted in Department of Computer Science and Engineering, Kalasalingam Academy of Research and Education(Affiliated to Deemed to be University) is an authentic record of my own work carried out during the period from Jan 2020 under the guidance of Mr.A.BHUVANESHWARAN. The work reported in this has not been submitted by me for the award of any other degree of this or any other institute.*

YAMSANI RAVI TEJA-9918004165

MADADI SAINATH REDDY-9918004162

### 3. ACKNOWLEDGEMENT

*First and foremost, I wish to thank the Almighty God for his grace and benediction to complete this Project work successfully. I would like to convey my special thanks from the bottom of my heart to my Parents and affectionate Family members for their honest support for completion of this Project. I express deep sense of gratitude to “Kalvivallal” Thiru.T. Kalasalingam B.com, Founder Chairman, “Ilayavallal” Dr.K.Sridharan Ph.D, Chancellor, Dr.S.ShasiAnand, Ph.D, VicePresident, Mr.S.ArjunKalasalingam M.S., VicePresident, Dr.R.Nagaraj Vice-Chancellor, Dr.V.Vasudevan Ph.D., Registrar , Dr.P.Deepalakshmi Ph.D., Dean (School of Computing) . And also a special thanks to Dr. A. FRANCIS SAVIOUR DEVARAJ. Head Department of CSE, Kalasalingam Academy of Research and Education for granting the permission and providing necessary facilities to carry out Project work. I would like to express my special appreciation and profound thanks to my enthusiastic project Supervisor Mr. A.BHUVANESHWARAN for his inspiring guidance, constant encouragement with my work during all stages. I am extremely glad that I had a chance to do my Project under my Guide, who truly practices and appreciates deep thinking. And during the most difficult times when writing this report, he gave me the moral support and the freedom I needed to move on.*

**YAMSANI RAVI TEJA-9918004165**

**MADADI SAINATH REDDY-9918004162**

## 4. TABLE OF CONTENTS

### 4.1 chapter 1 INTRODUCTION

*Optical character Recognition (OCR) is a conversion of scanned or printed text images [1], handwritten text into editable text for further processing. This technology allows machine to recognize the text automatically. It is like combination of eye and mind of human body. An eye can view the text from the images but actually the brain processes as well as interprets that extracted text read by eye. In development of computerized OCR system, few problems can occur. First: there is very little visible difference between some letters and digits for computers to understand. For example it might be difficult for the computer to differentiate between digit “0” and letter “o”. Second: It might be very difficult to extract text, which is embedded in very dark background or printed on other words or graphics. In 1955, the first commercial system was installed at the reader’s digest, which used OCR to input sales report into a computer and then after OCR method has become very helpful in computerizing the physical office documents. There are many applications of OCR, which includes: License plate recognition (2,3,4,5,6,7,8,9) image text extraction from natural scene images, extracting text from scanned documents etc. The system proposed in [12] is to rectify the text retrieved from camera captured images. An OCR system proposed by Thomas Deselaers et al. is used for recognizing handwritten characters and converting these characters into digital text. A system presented by Apurva A. Desai is used to recognize Gujarati handwritten numeral by using Artificial Neural Network (ANN)*

## 4.2 chapter 2 PROJECT DESCRIPTION

### 4.2.0.1 HISTORY OF TESSERACT

*Tesseract is an open source optical character recognition engine [7]. It was developed at HP in between 1984 to 1994. It was modified and improved in 1995 with greater accuracy. In late 2005, HP released Tesseract for open source. It is now available. It is highly portable. It is more focused towards providing less rejection than accuracy. Currently only command base version is available. As of now Tesseract version 3.01 is released and available for use. HP never used it. Now it is developed and maintained by Google. It provides support for various languages.*

### 4.2.1 ARCHITECTURE OF TESSERACT

*Tesseract OCR works in step by step manner as per the block diagram shown in fig. 1. First step is Adaptive Thresholding, which converts the image into binary images. Next step is connected component analysis, which is used to extract character outlines. This method is very useful because it does the OCR of image with white text and black background. Tesseract was probably first to provide this kind of processing. Then after, the outlines are converted into Blobs. Blobs are organized into text lines, and the lines and regions are analyzed for some fixed area or equivalent text size. Text is divided into words using definite spaces and fuzzy spaces. Recognition of text is then started as two-pass Largeprocess as shown in fig 1. In the first pass, an attempt is made to recognize each word from the text. Each word passed satisfactory is passed to an adaptive classifier as training data. The adaptive classifier tries to recognize text in more accurate manner. As adaptive classifier has received some training data it has learn something new so final phase is used to resolve various issues and to extract text from images. More details regarding every phase are available.*

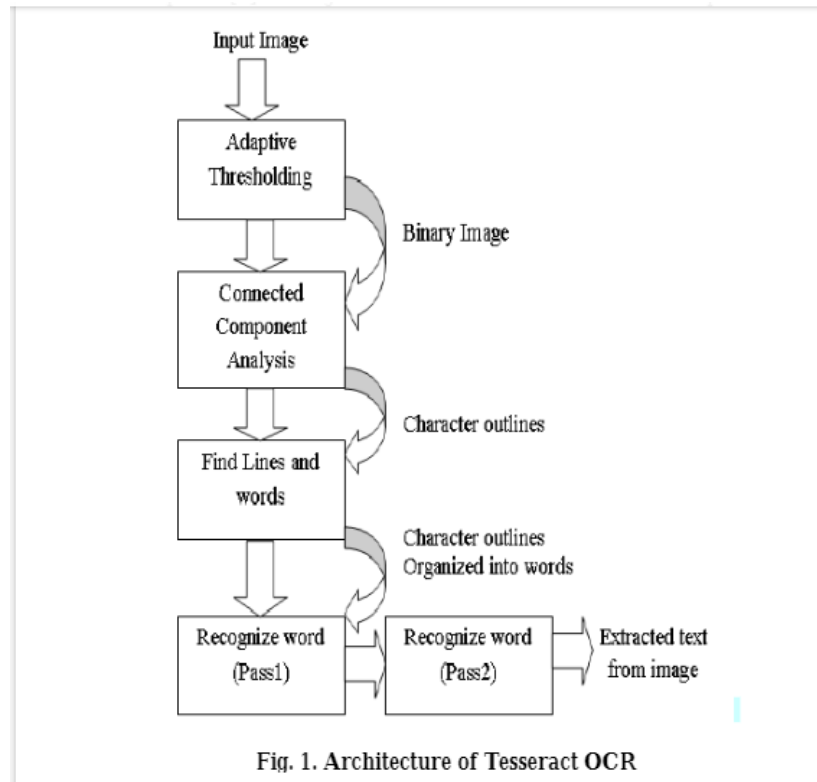


Fig. 1. Architecture of Tesseract OCR

### 4.3 WORKING

An image with the text is given as input to the Tesseract engine that is command based tool. The image is shown in fig . Then it is processed by Tesseract command . Tesseract command takes two arguments: First argument is image file name that contains text and second argument is output text file in which, extracted text is stored. The output file extension is given as .txt by Tesseract, so no need to specify the file extension while specifying the output file name as a second argument in Tesseract command.

```

Administrator: Command Prompt
Microsoft Windows [Version 10.0.18363.778]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd c:\

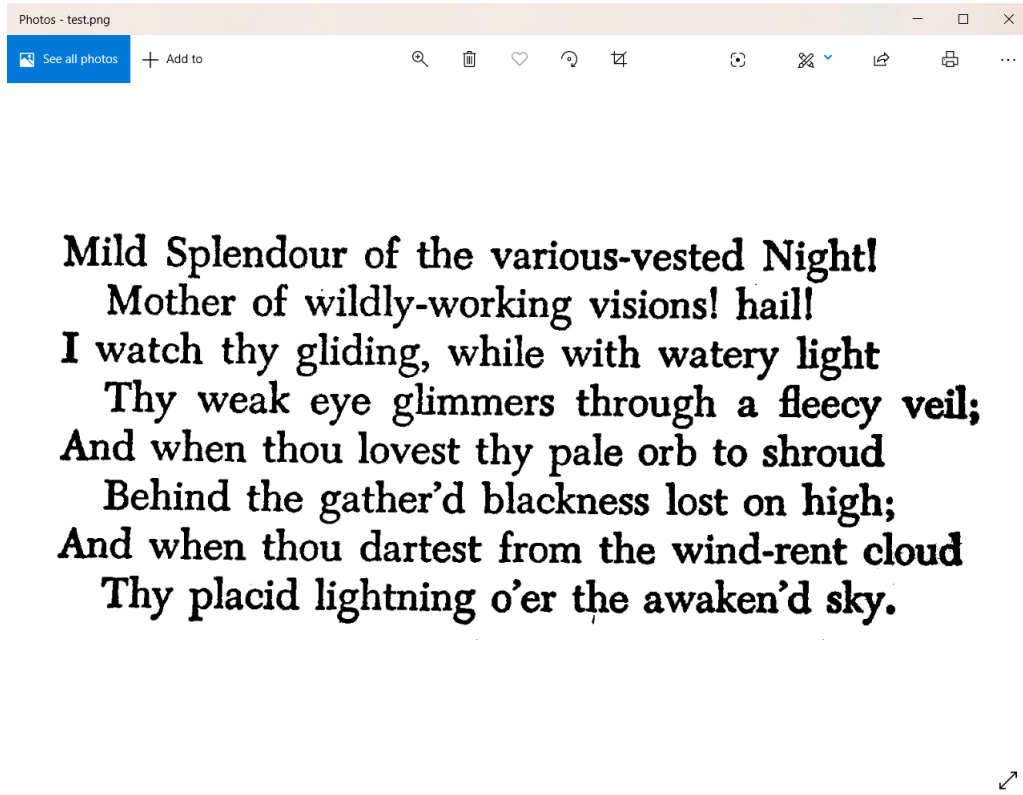
c:\>cd "Program Files"

c:\Program Files>cd Tesseract-Ocr

c:\Program Files\Tesseract-OCR>Tesseract test.png ravi.txt
Tesseract Open Source OCR Engine v5.0.0-alpha.20200328 with Leptonica
  
```

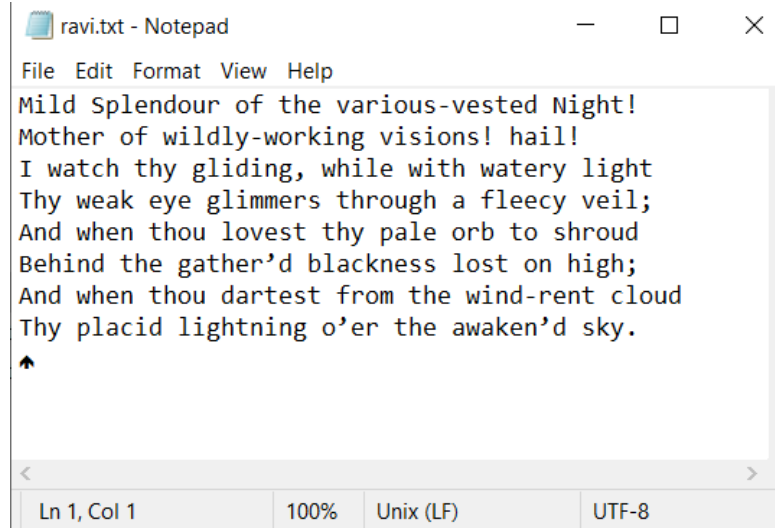


*As Tesseract supports various languages, the language training data file must be kept in the tessdata folder. In this research, the purpose is to extract English text from the images so we have kept only English language file in the tessdata folder After processing is completed, the content of the output file shown in fig. In simple images with or without color (gray scale), Tesseract provides results with best accuracy*



*Input Image in the .png format is*

*But in the case of some complex images Tesseract provides better accuracy results if the images are in the gray scale mode as compared to color images. To prove this hypothesis, OCR of same color images and gray scale images is performed and in both cases different result are achieved.*

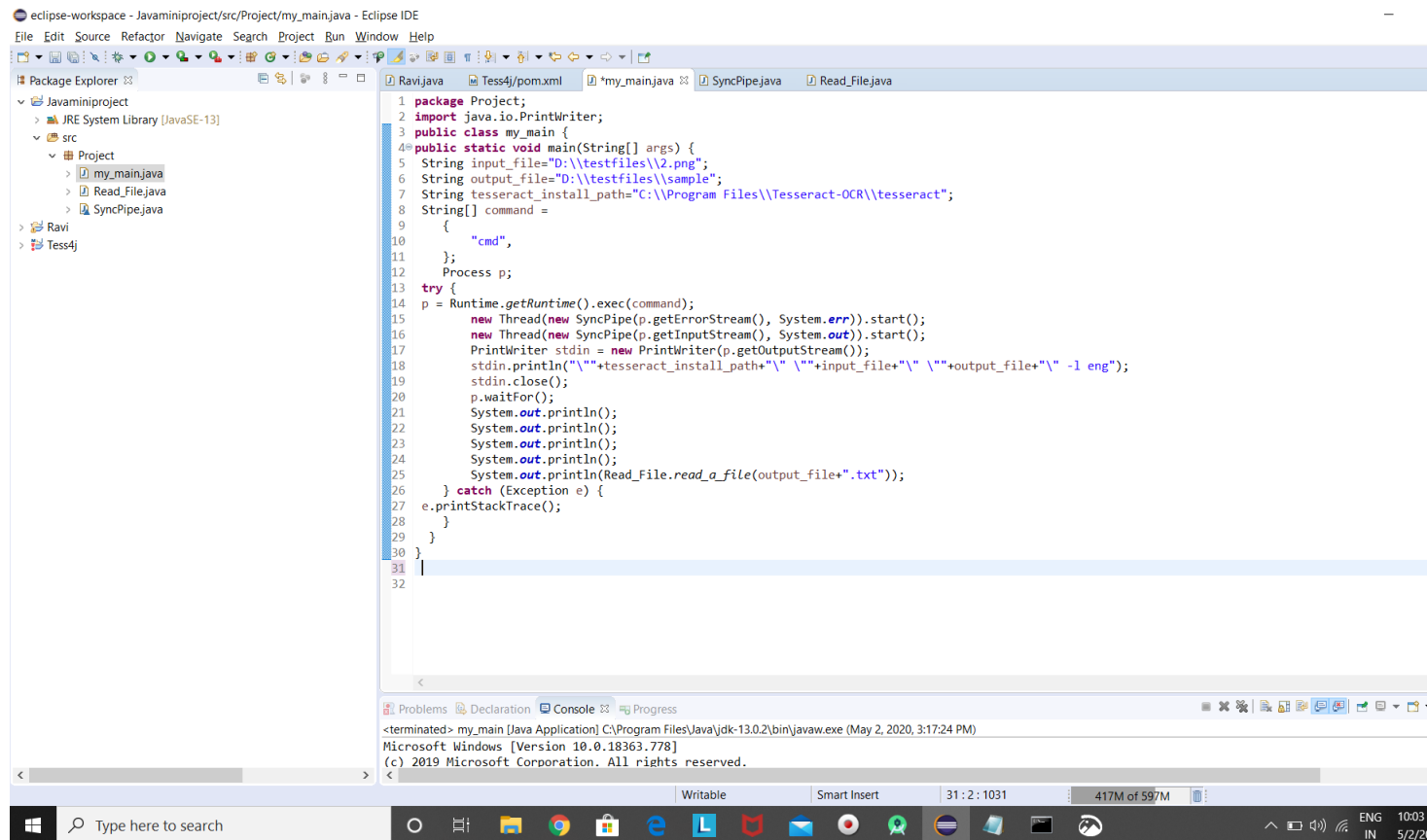


```
File Edit Format View Help

Mild Splendour of the various-vested Night!
Mother of wildly-working visions! hail!
I watch thy gliding, while with watery light
Thy weak eye glimmers through a fleecy veil;
And when thou lovest thy pale orb to shroud
Behind the gather'd blackness lost on high;
And when thou dartest from the wind-rent cloud
Thy placid lightning o'er the awaken'd sky.

Ln 1, Col 1    100%    Unix (LF)    UTF-8
```

## CODE OF THE PROJECT



```
eclipse-workspace - Javaminiproject/src/Project/my_main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer
Javaminiproject
  JRE System Library [JavaSE-13]
  src
    Project
      my_main.java
      Read_File.java
      SyncPipe.java
    Ravi
    Tess4j

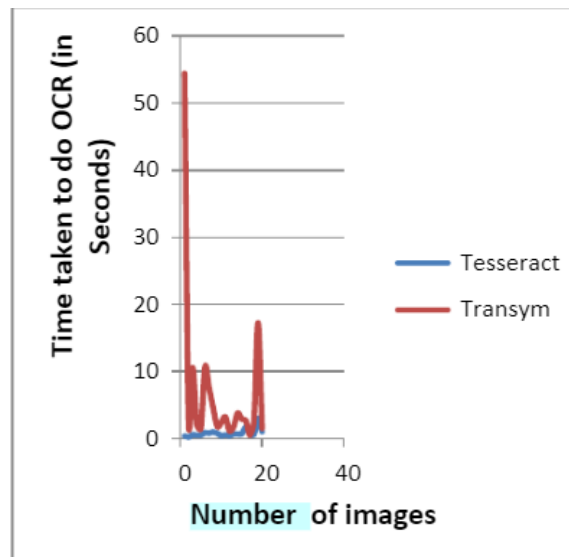
1 package Project;
2 import java.io.PrintWriter;
3 public class my_main {
4     public static void main(String[] args) {
5         String input_file="D:\\testfiles\\2.png";
6         String output_file="D:\\testfiles\\sample";
7         String tesseraect_install_path="C:\\Program Files\\Tesseract-OCR\\tesseract";
8         String[] command =
9             {
10                 "cmd",
11             };
12         Process p;
13         try {
14             p = Runtime.getRuntime().exec(command);
15             new Thread(new SyncPipe(p.getErrorStream(), System.err)).start();
16             new Thread(new SyncPipe(p.getInputStream(), System.out)).start();
17             PrintWriter stdin = new PrintWriter(p.getOutputStream());
18             stdin.println(" "+tesseraect_install_path+"\\ "+input_file+"\\ "+output_file+"\\ -l eng");
19             stdin.close();
20             p.waitFor();
21             System.out.println();
22             System.out.println();
23             System.out.println();
24             System.out.println();
25             System.out.println(Read_File.read_a_file(output_file+".txt"));
26         } catch (Exception e) {
27             e.printStackTrace();
28         }
29     }
30 }
31
32

Problems Declaration Console Progress
<terminated> my_main [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (May 2, 2020, 3:17:24 PM)
Microsoft Windows [Version 10.0.18363.778]
(c) 2019 Microsoft Corporation. All rights reserved.
```

*Here we can see 3 modules of code i.e(my main, syncpipe, Read file*

## 4.4 CONCLUSIONS

*Although Tesseract is command-based tool but as it is open source and it is available in the form of Dynamic Link Library, it can be easily made available in graphics mode. The results obtained in above sections are obtained by extracting vehicle number from vehicle number plate. So above results do not confirm that Tesseract is always better or faster than Transym but is it more accurate in extracting text from the vehicle number plate. The input images are specific, which are vehicle number plates, so in these specific images Tesseract provides better accuracy and in other kinds on images Transym might provide better accuracy than Tesseract. As we are interested in extracting vehicle number from vehicle number plate, we have considered both tools for serving this specific purpose.*



## 5. REFERENCES

*ARCHANA A. SHINDE, D. 2012.Text  
Pre-processing and Text Segmentation for OCR.  
International Journal of Computer Science  
Engineering and Technology, pp. 810-812.*

*ANAGNOSTOPOULOS,C.,ANAGNOSTOPOULOS,  
I., LOUMOS, V, KAYAFAS, E. 2006. A License  
Plate Recognition Algorithm for Intelligent  
Transportation System Applications., IEEE  
Transactions on Intelligent Transportation Systems,  
pp. 377- 399*