
# Optical Character Recognition System

**Instructor:** Dr. Hamid Azzo

**CSCI 595: Research Literature & Techniques**

Ben Niu

# 1. Introduction

## *1.1 Project Description*

This project involves optical character recognition system (OCR). The rapid development of information technology enabled more and more resources and services to be acquired by a large amount of users. However, some printed useful information is not possible to edit and archive, such as old-times papers, invoices, forms and so on. It consumes substantial amount of time and manpower to type into a computer and re-edit [1].

"Optical Character Recognition, or OCR, is a technology that enables you to convert different types of documents, such as scanned paper documents, PDF files or images captured by a digital camera into editable and searchable data" [2]. The OCR system will become a useful tool for storing information in a convenient way.

OCR system is related to current popular discipline such as computer graphic, machine learning, artificial intelligence etc. However, it has strong correlation with computer graphic. Because processing on optical character is equivalent to image of each character.

The OCR can be implemented in tow types on-line and off-line. Online OCR refers to automatic conversion of text as it is written. It is also called "real-time character recognition". Offline OCR refers to static representation of handwriting as with recognizing a scanned image/PDF. [3] This project will only concentrate on offline OCR system due to less workload and limited device. Also, it can be further extended to printed characters and handwritten characters recognition. [4]

## *1.2 Literature Survey*

There are many existing OCR system or software that perform well on text-based English-alphabets. However, these systems or software fail to perform on some irregular characters. For instance, handwritten scripts and a well-known term– CAPTCHA. CAPTCHA stands for "Completely Automated Public Turing Test to Tell Computers and Humans Apart", and a "CAPTCHA is a program that generates and grades tests that most human can pass, but current computer programs cannot pass" [5].

Although, OCR system is widely used and play an effective role in many field of research presently, its accuracy still become a challenge today. Many OCR software or systems have common disadvantages affect rate of accuracy such like noisy characters, overlap, tilt, etc. There still have room for improvement. [6]

## *1.3 Proposed Solutions*

This project focus on text-based CAPTCHA; since it's the most commonly used and easy to implement at a low cost [7]. My plan is to develop an OCR system not only that can identify text-based English-alphabets, but also recognize handwritten scripts and CAPTCHA image samples [8]. For OCR system, Python is a powerful language that has a large number of libraries can be used for image processing. In this project we will focus on PIL (Python Imaging Library) and Tesseract. Tesseract is an OCR library that sponsored by Google; also it can be trained to recognize any number of fonts [9]. Therefore, PIL take over the first phase preprocessing; Tesseract handles the second phase recognition that will be

introduced as follows. After processing of OCR system, the original files will be converted into text.

The OCR system implementation consist of two major phases: 1) Preprocessing; 2) Recognition. For preprocessing, it has different algorithms performed on the scanned image depends on some factors such as paper quality, resolution of image, format and layout and also type of character: printed or handwritten. Typically, preprocessing includes following stages: noise removing, binarization, and segmentation. For recognition, usually it has two major stages: feature extraction and classification. Based on the procedure above the printed image like CAPTCHA or handwritten scripts will be converted into editable text. [6]

## *1.4 Proposed Features*

This project will implement an OCR system that offers the following features:

1- Files import.

2- Recognition.

3- Export.

Opening and importing files is the initial step of the OCR system. Then, they can be loaded from disk or memory. Afterward files are loaded in the OCR system, the recognition starts dealing with corresponding files. In this stage the system can recognize individual character or a set of characters. Finally, after recognition the system will print characters into text.

# 2. Overall Architecture

## *2.1 Architecture Diagram*

**Figure 2.1 depicts the overall architecture of the OCR system.**

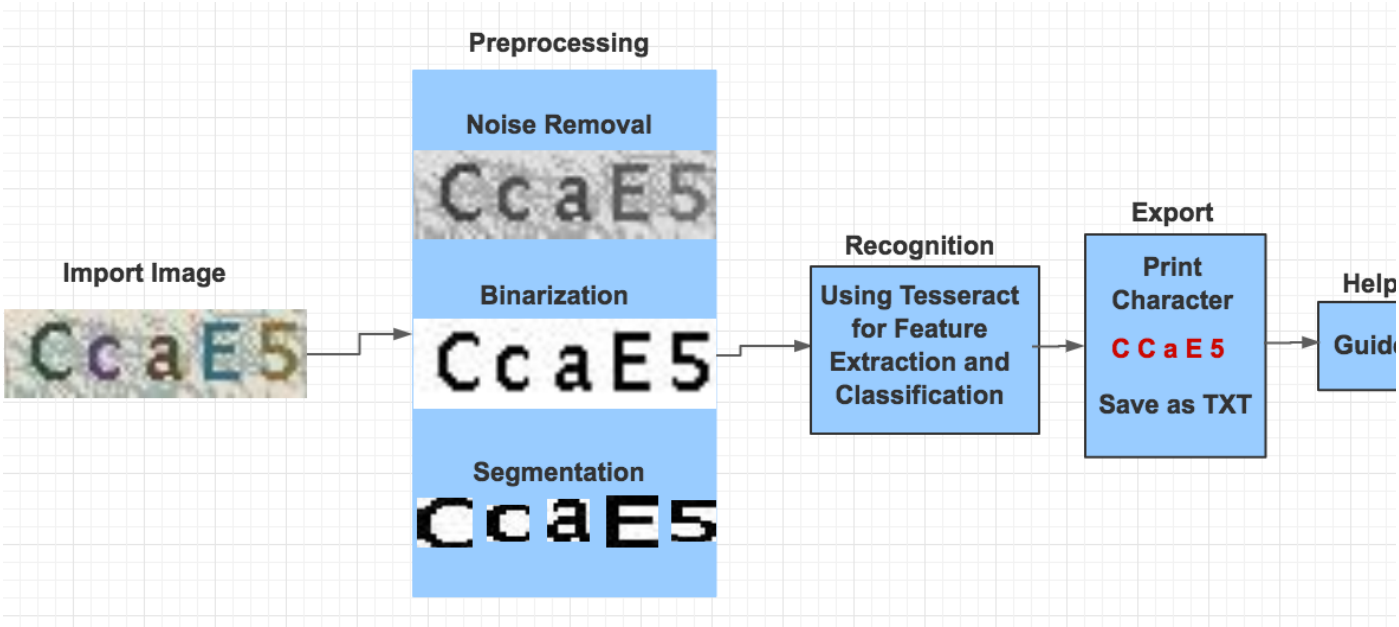



*Figure 2.1 Overall Architecture*

## *2.2 Architecture Descriptions*

**Import Image Module** – The import image is first module of all modules, it will be used for users to select image file from disk or memory, then import corresponding file to system. The import image module shall connect to the Preprocessing module so that a user may implement a series of processing on the image.

**Preprocessing Module** – The preprocessing module will provide a serial of processing that include the following modules: Noise Removal module,

Binarization module, and Segmentation module. In order to efficiently identify the image these modules are very significant for the next recognition.

**Noise Removal Module** – The noise removal module will provide a method to remove the noise that is random variation of brightness or color in an image, and it can make the character of image more difficult to read.

**Binarization Module** – The binarization module will provide a method to transform the grey scale image into a binary format. Because of uneven darkness of image background will give disadvantageous impacts on the results.

**Segmentation Module** – The segmentation module will provide a method to segment the letters into isolated characters that are recognized individually.

**Recognition Module** – The recognition module will be use Tesseract for feature extraction and classification. "The objective of feature extraction is to capture the essential characteristics of the symbols. The classification is the process of identifying each character and assigning to is the correct character class." It will be utilized for users to recognize corresponding image that imported and processed from above modules. The recognition shall connect to the export module so that a user may get the results printed or save as text file.

**Export Module** – The export module will be used for users to receive the results in different way. Not only user can edit characters directly, but also get TXT file that include corresponding characters.

**Help Module** – The help module will be used for users to see system guidelines and some common questions.

# 3. Activity Diagrams

There are six activity diagrams that capture all the activities performed within the OCR System.

## 3.1 OCR System Initializing

On system open-up the OCR System will perform file selection. It will open PC folder and select image file from disk or memory. Then import corresponding file to system.

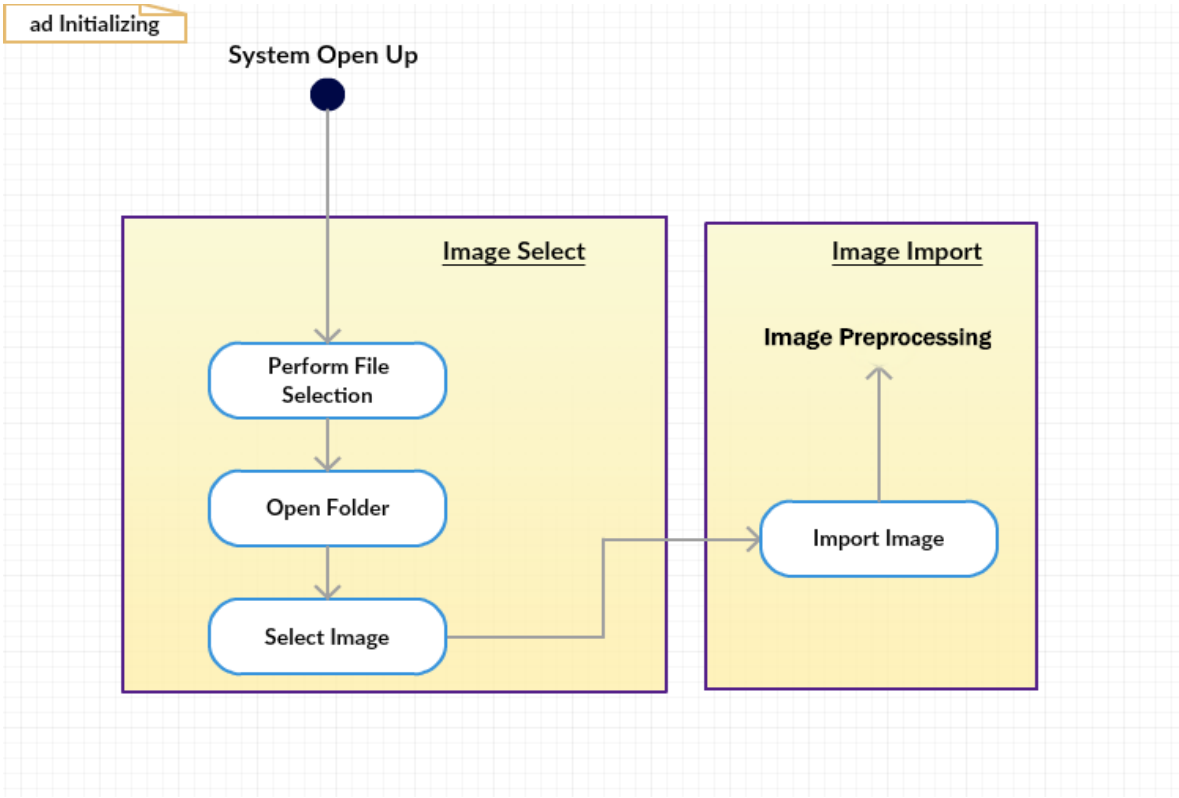



*Figure 3.1: Initializing Activity Diagram*

## 3.2 Character Writing

Except image import, the OCR system also offers a function of character writing. User will be able to write some letters or numbers on the canvas.

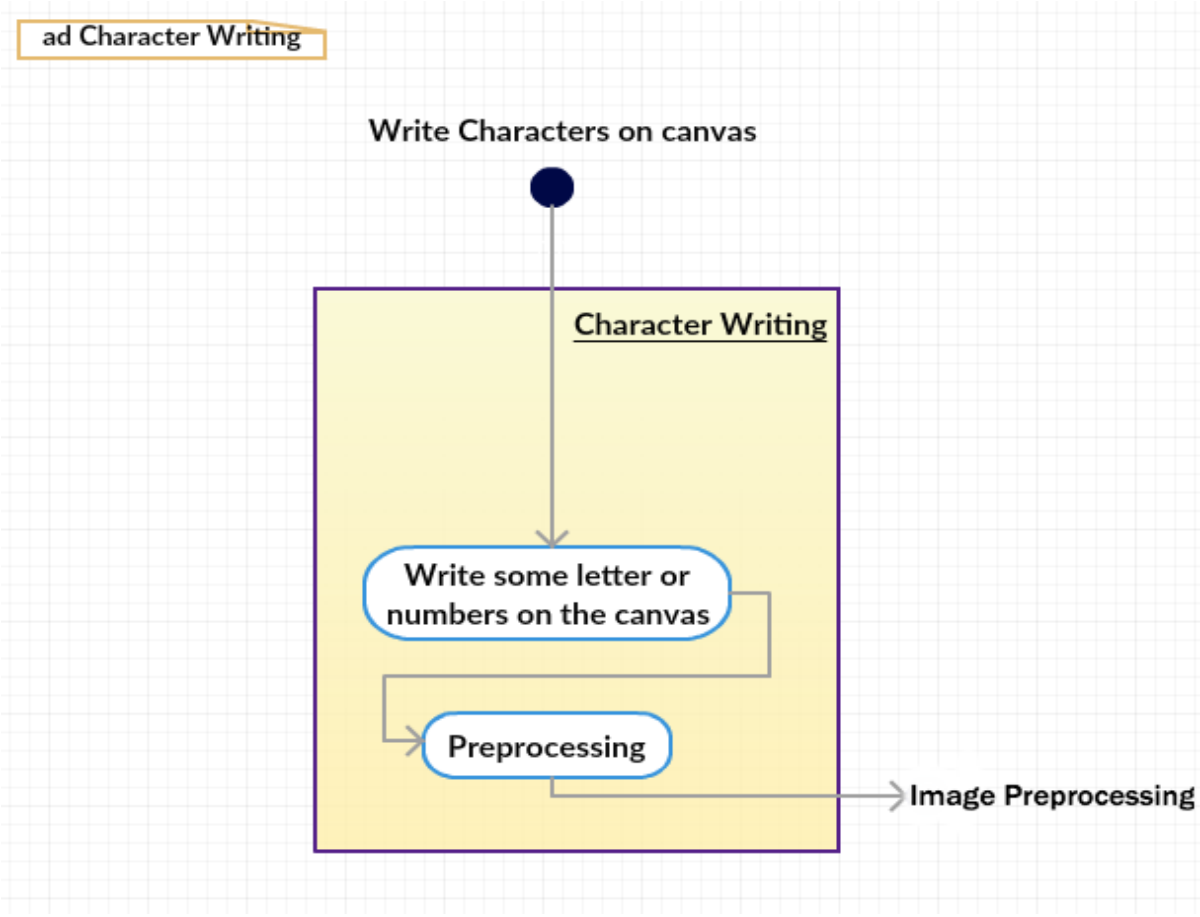



*Figure 3.2: Character Writing Activity Diagram*

## 3.3 Image Preprocessing

Once the image is imported, the preprocessing will run a serial of processing that include: Noise Removal, Binarization, and Segmentation. However, some pure gray images do not have to perform Noise Removal. After the preprocessing, the image file will be ready for recognizing.

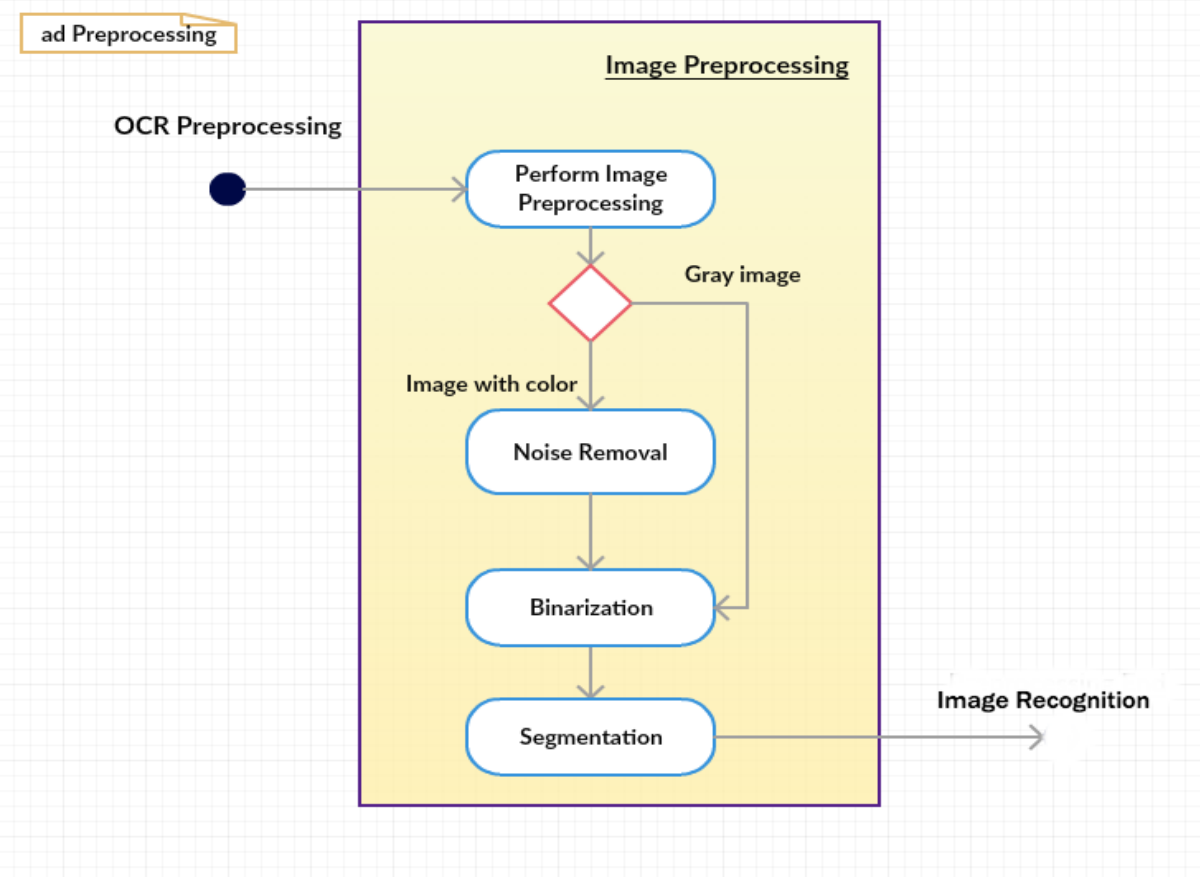



*Figure 3.3: Image Preprocessing Activity Diagram*

## *3.4 Image Recognition*

Once the image is preprocessed, the next step recognition is the most important part in the OCR System. It will use Tesseract for feature extraction and classification, and then recognize the corresponding image.

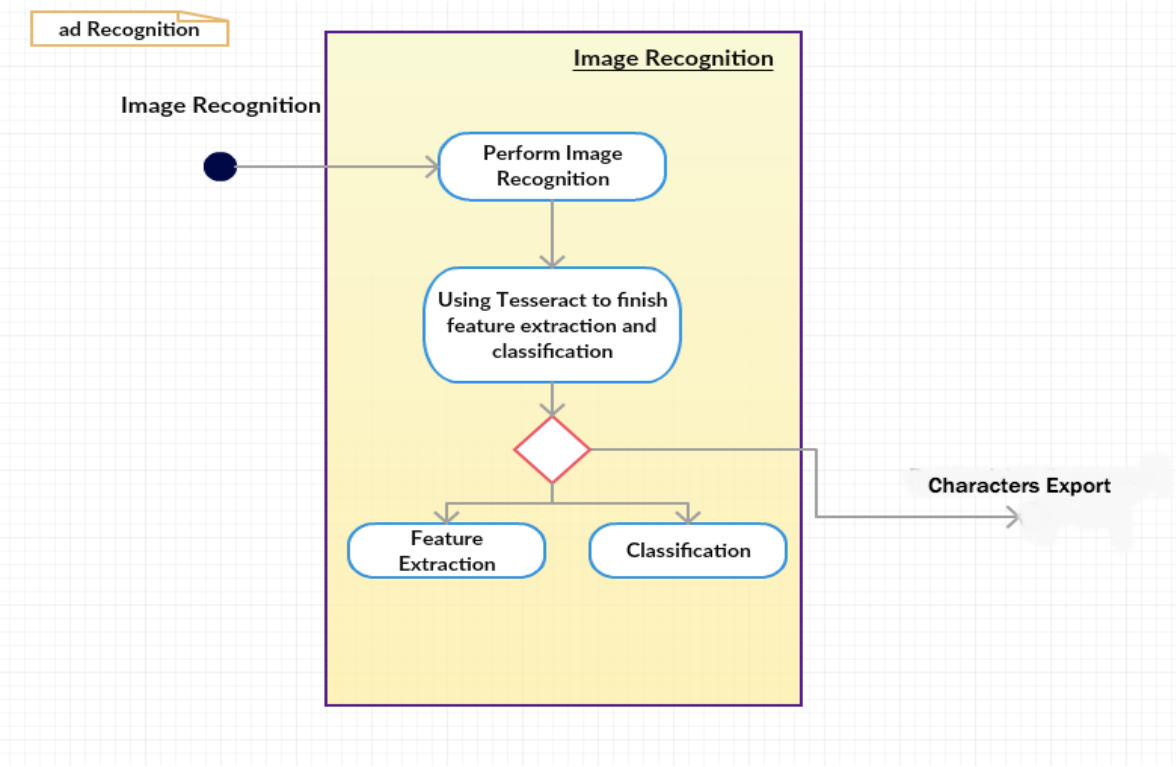



*Figure 3.4: Image Recognition Activity Diagram*

## *3.5 Character Export*

After recognition, the user may receive the results either directly or get text file that include corresponding characters.

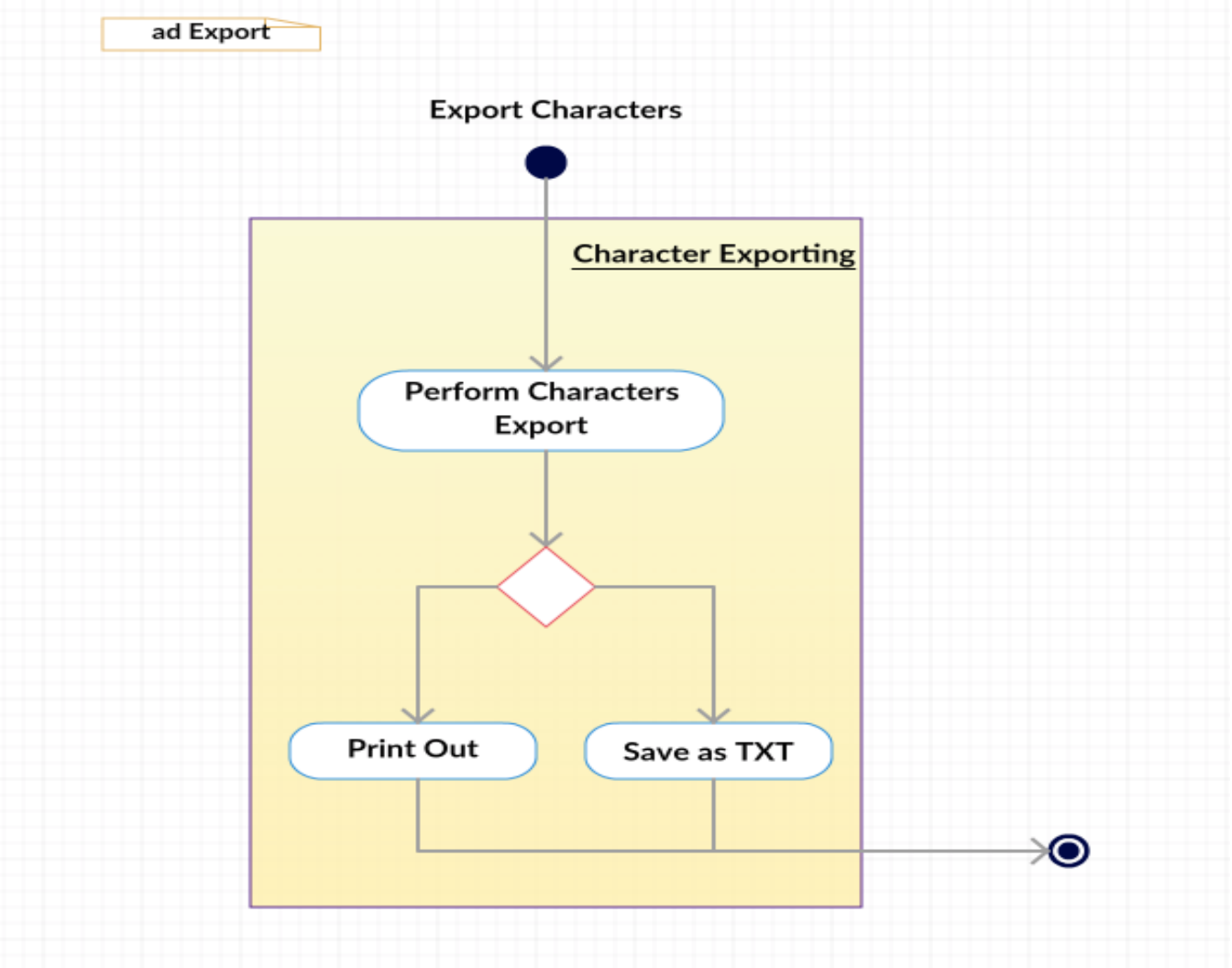



*Figure 3.5: Character Export Activity Diagram*

## *3.6 System Help*

The User may see system guidelines and common questions.

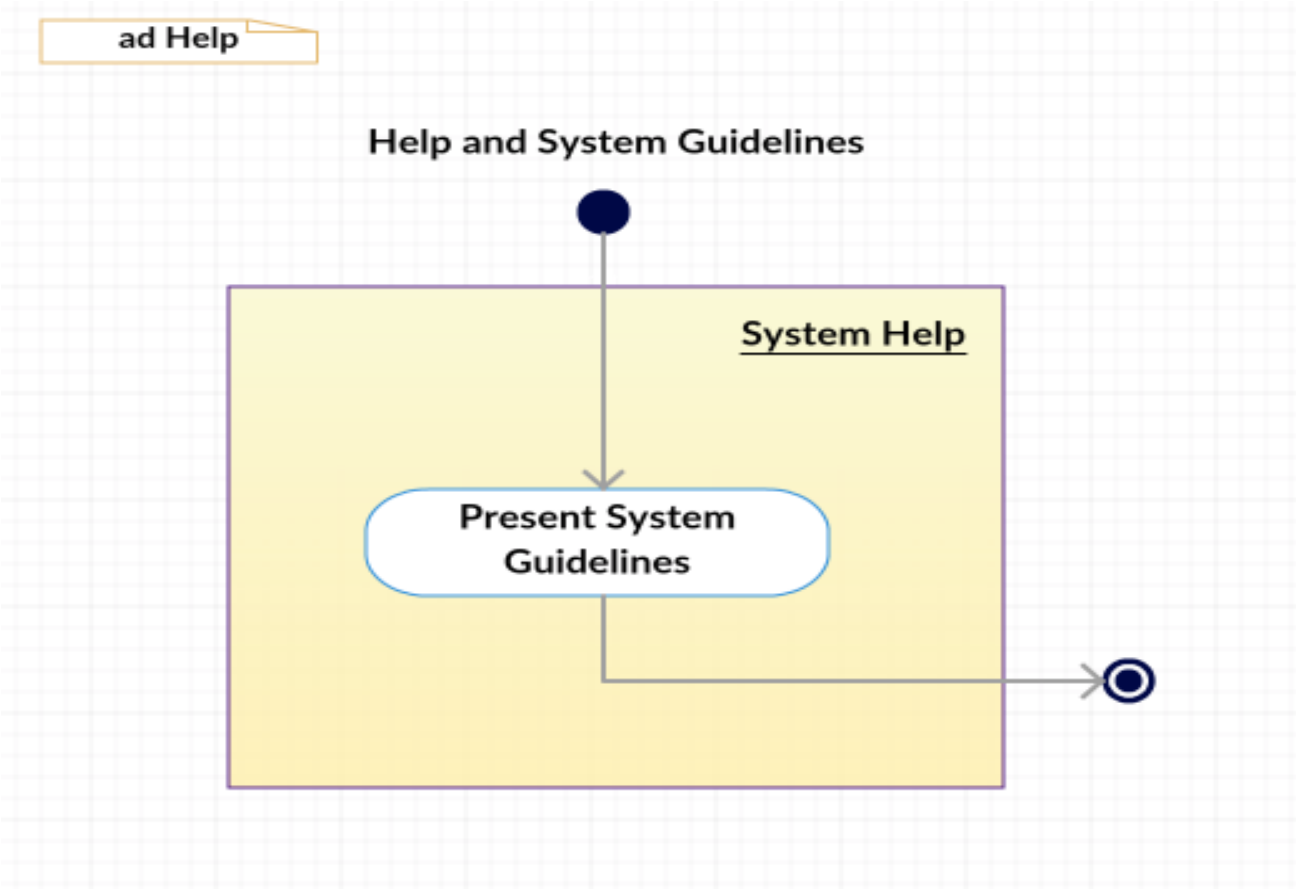
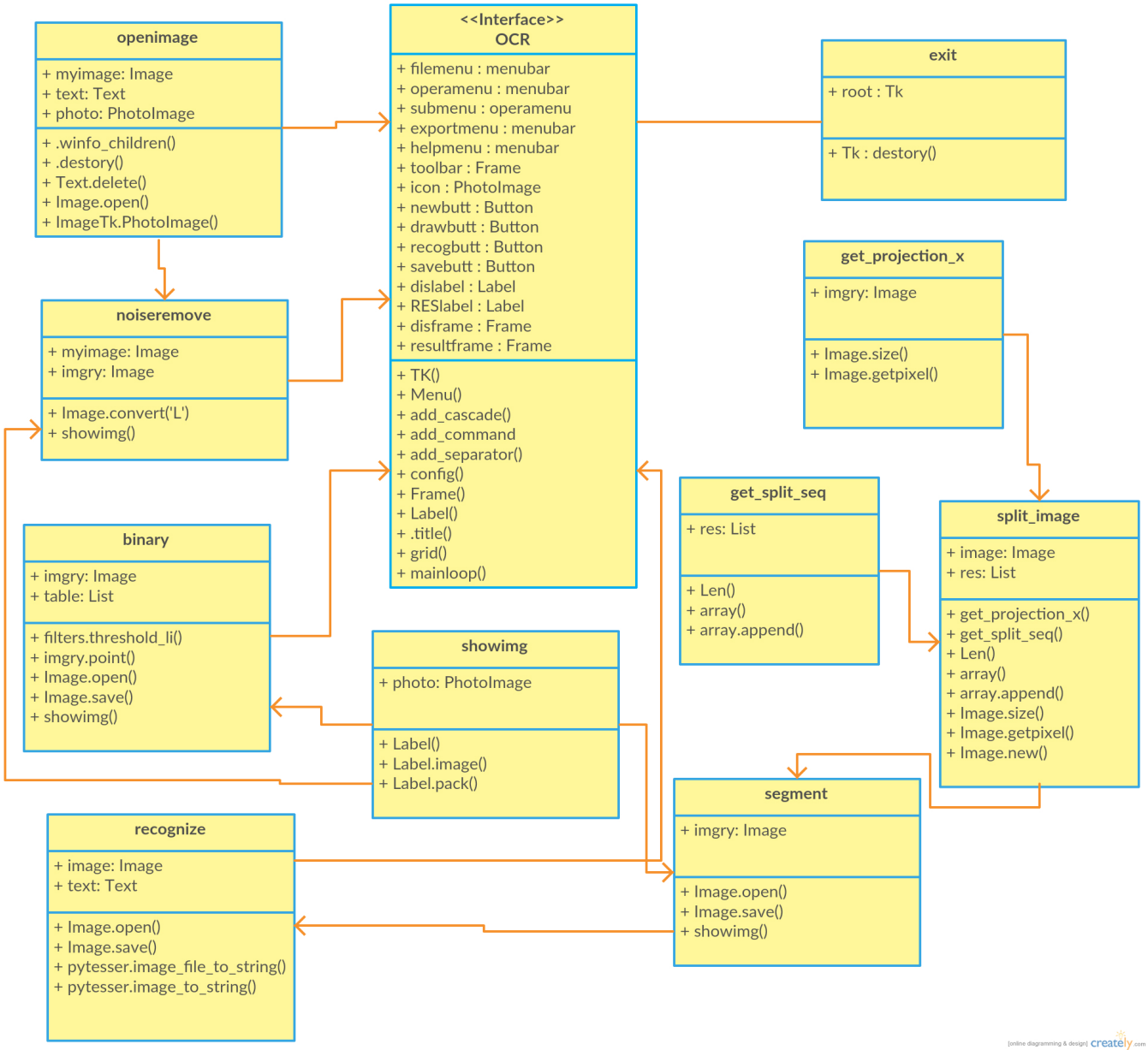



*Figure 3.6: System Help Activity Diagram*

# 4. Class Diagrams



# 5. Technical Requirement

***Software Used:***

| | | |
|---|---|---|
| Operating System | : | MacOS Sierra 10.12.1 |
| User Interface | : | Tkinter |
| Programming Language | : | Python 3.5 |
| IDE/Workbench | : | Anaconda- notebook 4.1.0 |
| Other | : | Enthought Canopy |

***Hardware Used:***

| | | |
|---|---|---|
| Memory | : | 256 GB |
| RAM | : | 8.00 GB |
| Processor | : | Intel(R) Core(TM) i5 CPU @ 2.4 GHz |

# 6. Conclusion

Through the OCR system, users can easily convert different types of documents into editable text. It will save a lot time and money for those who need some scanned documents turn to be editable text.

Also, this OCR system is easy to use and perform. The OCR system or software will become more and more popular in the near future.

# Functional Requirement

## Abbreviation

- **OCR** – Optical Character Recognition

**Objective**: This project is to implement an OCR system that can easily convert different types of documents into editable text. My goal is to save substantial amount of time and manpower by performing this effective system.

Optical Character Recognition System Functions

- File import–Select file from disk or memory, and then import it to system.
- Preprocessing–Perform different algorithms on selected file.
- Recognition–Start to recognize corresponding files via processing in system.
- Export – Print the characters into text, then it can be edit.

## Functional Requirements:

| S.No | Requirement ID | Requirement Description |
|---|---|---|
| 1 | OCR_REQ_1.1 | OCR shall allow user to select file from disk or memory. |
| 2 | OCR_REQ_1.2 | OCR shall allow user to select type of PDF or image file. |
| 3 | OCR_REQ_1.3 | OCR shall allow user to import file to system. |
| 4 | OCR_REQ_2.1 | OCR shall allow user to preprocess corresponding file. |
| 5 | OCR_REQ_2.2 | OCR shall allow user to remove noise of corresponding file. |
| 6 | OCR_REQ_2.3 | OCR shall allow user to convert the corresponding pixel file to a binary file. |
| 7 | OCR_REQ_2.4 | OCR shall allow user to get segmentation of corresponding file. |
| 8 | OCR_REQ_3.1 | OCR shall allow user to recognize corresponding file. |
| 9 | OCR_REQ_3.2 | OCR shall allow user to get feature extraction of corresponding |

| | | |
|---|---|---|
| | | file. |
| 10 | OCR_REQ_3.3 | OCR shall allow user to get classification of corresponding file. |
| 11 | OCR_REQ_4.1 | OCR shall allow user to export corresponding file. |
| 12 | OCR_REQ_4.2 | OCR shall allow user to print out characters that equivalent to file. |
| 13 | OCR_REQ_4.3 | OCR shall allow user to save results as .txt file. |
| 14 | OCR_REQ_5.1 | OCR shall allow user to get help when they encounter problem. |
| 15 | OCR_REQ_5.2 | OCR shall allow user to view the guide for the system. |

## References:


1. Najib Ali Mohamed Isheawy and Habibul Hasan. "Optical Character Recognition (OCR) System", IOSR-JCE, Volume 17, Issue 2, Ver. II (Mar – Apr. 2015), PP 22-26.
2. ABBYY. (2016). What is OCR and OCR technology. Retrieved September 13, 2016, from https://www.abbyy.com/finereader/about-ocr/what-is-ocr/
3. Optical character recognition (2016). . In *Wikipedia*. Retrieved from https://en.wikipedia.org/wiki/Optical_character_recognition
4. Pai, N., & Kolkure, V. S. (n.d.). Classification and Comparative Analysis of Character Recognition Techniques. Asian Journal of Convergence in Technology, 1(5),
5. L Von Ahn, M Blum and J Langford. "Telling Humans and Computer Apart Automatically", CACM, V47, No2, 2004.
6. Shodhganga. A reservoir of Indian Theses. Retrieved September 16, 2016, from http://shodhganga.inflibnet.ac.in/bitstream/10603/4166/10/10_chapter%202.pdf
7. Kiranjot Kaur*, Sunny Behal, "Designing a Secure Text-Based CAPTCHA", Procedia Computer Science 57 (2015) 122 – 125.

8. M. A. Asim K. Jalwana, Muhammad Murtaza Khan, Muhammad U. Ilyas. “Automatic Identification of CAPTCHA Schemes”, 10th International Symposium, ISVC 2014.
9. Mitchell, R. (2015). *Web scraping with python: Collecting data from the modern web.* O’Reilly Media.