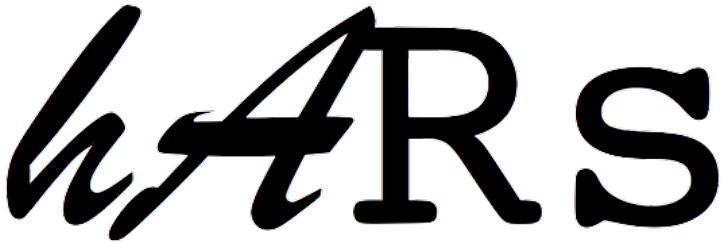
HARS

(Handwritten Annotation Removal System)



Group-8

Anuj Kr. Pathak<anujpathak9335@gmail.com>

Sayan Shankhari<sayanthecomputerguy@gmail.com>

https://github.com/TheScienceUniverse/ComSoft/tree/master/HARS

June 4, 2018

Contents

1 Introduction **1**

2 Project Proposal **3**
2.1 Task-1 3
2.1.1 Conditions 3
2.1.2 Procedure 8
2.2 Task-2 8
2.2.1 Conditions 8
2.2.2 Procedure 9

Abstract

An **OCR**(Optical Character Recognizer) is a system to recognize the text from a scanned document. The performance of an OCR is reduced very much when the document is having unwanted or irrelevant writings or any means of markations like handwritten annotations. To remove annotations, we can use **HARS**(Handwritten Annotation Removal System) to gain performance of OCR to about 100%. Many documents having both historical and aesthetic values are often found full of handwritten annotations for either correction by reader or for reading and remembering purpose. Therefore, our target of this research project will be to remove the handwritten annotation in such a way that OCR have not to face problems to recover the text, even the document would be clean and clear to normal Human reader. In this research, we will consider between-the-line and between-the-letters annotation removing but also trial to remove annotation from each part of the document. In the first part we will take plaintext in black-&-white format, after we'll consider coloured documents.

Chapter 1

Introduction

Annotation to a computer printed document refers to the process of adding or writing critical commentary or exaplanation or even correction at the different parts of the document. The annotations are sometime important observations or correction, that's why it's very challenging to remove them while the original document is not available at all.

A reader can add different types of annottations of different parts of the document. For example, the annotations can be underscore lines, side scoring to highlight keywords, sentences, enclosing circles, brackets, or crosses, strike-through or even hand-drawn images. The annotations can be between the lines, or between letters or boundery region out of text section.

These type of works are done by very few scholars earlier. Like,

- **Mori and Bunke** successfully removed coloured or unstructured annotations,
- **Guo and Ma** proposed method is for separation of **handwritten annotations** from **machine printed document**,
- **Y. Zheng** *et. al.* proposed method to segment and identify the two from noisy documents,
- **Lincoln Faria da Silva** *et al.* proposed a method for same work in a scanned document image,
- New features for distinguishing and removing the handwritten text and machine printed text are also proposed
  - Vertical Projection Variance,
  - Major Horizontal Projection Difference,
  - Pixels Distribution,
  - Vertical Edges,
  - Major Vertical Edge.

  Unfortunately, their method can work only when the machine printed text and handwritten texts are separated.

However, they have limitations to extract correctly and moreover some of them are doing job of OCR, thus double effort.

So, in this project, we'll start making HARS by feeding plaintext Black-N-White document having handwritten annotations and begin to remove the between-the-lines and between-the-letters annotations using different techniques like pixel analysis, connected component detection, segmentation, statistics and prediction, and later we'll go for coloured documents and try to invent more general as well as more advanced method for any text document.

Chapter 2

Project Proposal

2.1 Task-1

The first task would be working on Black-N-White soft (digital) document.

2.1.1 Conditions

Input: Document having(/without having) handwritten annotations on it.
Output: Document without having annotations on it. Input Conditions are as follows,

- **File-Type:** .pdf (Scanned)
- **Page-Color:** White/Light-Gray
- **Font-Color:** Black/Dark-Gray
- **Annotation Color:** Any color other than black

Explanation

The causes of restrictions of the input format are the following,

- Scanned pdf file
  As the file format of PDF (Portable Document Format) is mostly accepted by most of the popular digital vidual devices like Computers (drivers are available), Mobiles, Servers. PDF file format (Invented by Adobe, Reconfigured by ISO) is described in the following Reference, https://ia600106.us.archive.org/34/items/ListOfFileFormats/Files/PDF32000_2008.pdf
  The PDF file has collection of different objects like, text sections, links, images *etc.* That's why it is not so simple to decrypt. But after consulting the file format, we can easily access bit values of a pdf file and process it according to our need. And moreover if the Scanned document is taken, as it would be suitable conversation of .bmp (BitMaP image file) → .pdf, a single type of object would be there for process on bit values.
  We've not choosen any type of image file (like .jpg/.jpeg, .png, .bmp) as it would be difficult for the software and the software would be big as well as complex to handle. It is better to stick to universal pdf file.

- Page-Color is by default choosen by the very first pixel of the Input pdf document as there is no Legal document or Book that consists of Letter starts from 1st pixel, and the annotation provider has blocked the path with another color rather than White or Light-gray, the program will set it by White.
- There would be no font recognized by the Software. Rather than that, we'll take the color of a pixel in RGB (Red-Green-Blue) color format in Hexadecimal form ($00 \rightarrow$ FF) and if it is not in the preferred range, we'll set the color to Page-Color.
- As we are taking Black-N-White document, where font is black, if the color of the annotation is black, after scanning, there would be no difference between the color of font and annotation. As about 90% of the legal or archival documents are written in black font and more than 50% of them having annotation of not black color to be noticed by the reader next time while reading, it is suitable to take such documents.

Example: The next page is a sample original TeX to PDF document, and the next to next page is the scanned document of the former one.

Preamble to the Constitution of India

We, **THE PEOPLE OF INDIA**, having solemnly resolved to constitute India into a **SOVEREIGN DEMOCRATIC REPUBLIC** and to secure to all it's citizens:
JUSTICE, social, economic and political;
LIBERTY of thought, expression, belief, faith and worship;
EQUALITY of status and of opportunity;
and to promote among them all
FRATERNITY assuring the dignity of individual and the unity of the Nation;
IN OUR CONSTITUTION ASSEMBLY this twenty-sixth day of November, 1949, do **HEREBY ADOPT, ENACT AND GIVE TO OURSELVES THIS CONSTITUTION.**

Preamble to the Constitution of India

We, **THE PEOPLE OF INDIA**, having solemnly resolved to constitute India into a **SOVEREIGN DEMOCRATIC REPUBLIC** and to secure to all it's citizens:
JUSTICE, social, economic and political;
LIBERTY of thought, expression, belief, faith and worship;
EQUALITY of status and of opportunity;
and to promote among them all
FRATERNITY assuring the dignity of individual and the unity of the Nation;
IN OUR CONSTITUTION ASSEMBLY this twenty-sixth day of November, 1949, do **HEREBY ADOPT, ENACT AND GIVE TO OURSELVES THIS CONSTITUTION.**

composition

Preamble to the Constitution of India

Introduction

LARGEST & MOST COMPLEX

deep sincerity

We, **THE PEOPLE OF INDIA**, having solemnly ~~resolved to~~ constitute India into a **SOVEREIGN, DEMOCRATIC REPUBLIC** and to secure to all it's citizens: SOCIALIST SECULAR

JUSTICE, social, economic and ~~political~~;

LIBERTY of thought, expression, belief, faith and worship;

EQUALITY of status and of opportunity;

and to promote among them all

FRATERNITY assuring the dignity of individual and the unity of the Nation; quality of being worthy of honour

26/11/1949

IN OUR CONSTITUTION ASSEMBLY this twenty-sixth day of November, 1949, do **HEREBY ADOPT, ENACT AND GIVE TO OURSELVES THIS CONSTITUTION.**

legally choose to follow (make laws)

Sovereign :- Supreme power

Democratic :- Governed by whole population

Republic :- supreme power held by elected people.

Socialist :- practising socialism

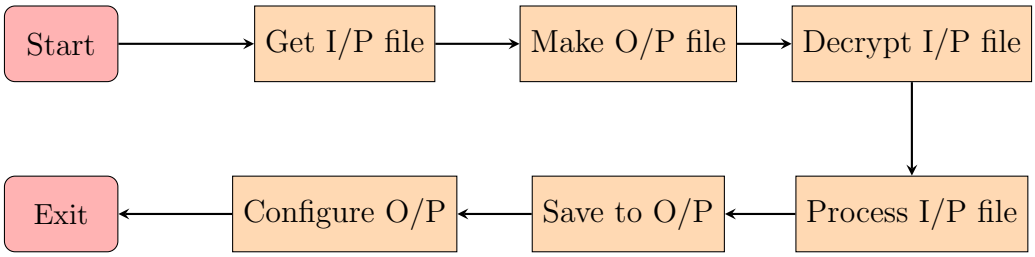
Secular :- not connected with religious / spiritual matter

Fraternity :- group of people, having / sharing common profession / interest

By seeing the previous pages you can get the summery of the idea.
1st one is original TeX generated .pdf file
2nd one is scanned .pdf file of original printed document
3rd one is scanned .pdf file of annotated printed document
The third file and files like this would be the input of the software and the expected output have to look like 1st file or atleast 2nd file.

2.1.2 Procedure

The procedure is as follows,



- **Get I/P:** There have to be a simple system in the software to accept the proper input file location choosen by the User.
- **Make O/P:** At first the output file should be created as .txt file.
- **Decrypt I/P:** As explained earlier, the input is in form of .pdf (converted from scanned .bmp), we should convert the .pdf to .bmp format to see visually what happens while processing.
- **Process I/P:** We can directly process on the input .pdf file, or created .bmp image file. The process is accessing the bit values
- **Save to O/P:** While processing either we can save data in direct .pdf format or the .bmp format.
- **Configure O/P:** After all these, we've to give final touch on the output, by either reconfiguring the output .pdf file or suitable convertion of .bmp to .pdf with having optional features like, Signature, Password, *etc.* The final output .pdf file location have to be given to the User.

2.2 Task-2

The 2nd and final task is working on coloured document.

2.2.1 Conditions

Input: Document having(/without having) handwritten annotations on it.
Output: Document without having annotations on it. Input Conditions are as follows,

- **File-Type:** .pdf (Scanned)
- **Page-Color:** Any Color

- **Font-Color:** Any Color
- **Annotation-Color:** Any Color

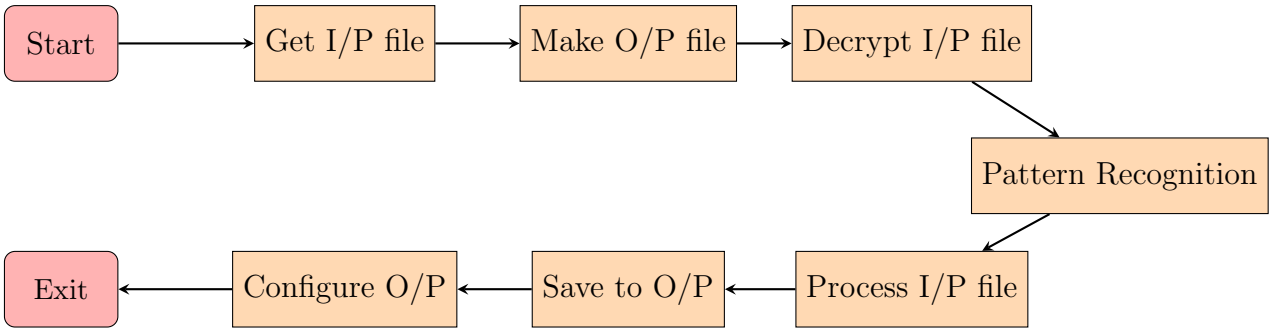
Explanation

The causes of restrictions of the input format are the following, *Scanned .pdf is discussed here →2.1.1*

- **Page Color:** Any color diferent than font color
- **Font-Color:** Any color diferent than font color
- **Annotation-Color:** Any Color

2.2.2 Procedure

The procedure is as follows,



- Same items are same as previous Task 2.1.2.
- Additional **Pattern Recognization** is the Heart or the Core idea of the success of our project. In this case, we'll have statistical analysis of the patterns of the letters and other punctuation marks of different types of font families, stored in the database. Instead of storing plenty of font drawing formats in the database, we can use vector formats (Complex but little storage size).

  The job of the pattern-recognizer will be recognize connected component (using well known flood-fill method in graphics) from the i/p image and give the location in the font database, if found, do nothing, & if not, mark the pixels for deletion *i.e.* recolour with page-color. The another feature of the recognizer would be recognize the width & sharpness of the edge of the components, as we know, no matter how much scaling we do to annotate with our hand we can never get the exact match with the sharpness and width of the computer printed text.