Generic PDF to Text Conversion using Machine Learning

Chandrahas Gaikwad
Department of Computer Engineering,
SITS, Pune

Satish Akolkar
Department of Computer Engineering,
SITS, Pune

Reshma Khodade
Department of Computer Engineering,
SITS, Pune

Deepali Dalal
Department of Computer Engineering,
SITS, Pune

Swarupa Kamble
Department of Computer Engineering,
SITS, Pune

ABSTRACT
The world is advancing to a futuristic paperless aeon. Stockpiling of logs, charters, records and other documents has become monotonous. Storage of these as 'soft copy' is more convenient and reliable. This facilitates searching and sorting with ease. Generally such documents are stored as PDF (Printable Document Format), so as to make the documents easily viewable and avoid unnecessary changes emerging due to software platforms. However, editing of locally scripted documents becomes inconvenient. The conventional PDF to text conversion software are incapable of editing some unexplored scripts. In this research paper, a generic way of making PDF documents editable by the script-independent and machine learning features is presented. This is possible by slicing out the characters from the PDF. A set of classifiers is applied to identify the character. The Decision Model implemented as a part of Machine learning systematizes the classifier functions. The resultant classifier set gives the resolution for the character. This approach eradicates the barrier of limiting our scope to international scripts and also facilitates usage of regional scripts in the technological world.

General Terms
Editing, slicing, documents, conversion

Keywords
Generic, Machine learning, script-independent

1. INTRODUCTION
The tremendous advancements in the technological era has led to flourishing of activities in different sectors. Owing to this, the maintenance of records for all these activities became crucial. In earlier days, every record was stored in handwritten paper format. Huge registers and catalogues thus came into picture. But as scope of development increased, accessing these records and their sorting became tedious. Later when information was digitized i.e. in the generation of computers, storing and accessing of data became much easier. Any part of the information could be made available at any point of time. Generally information was stored in the form of .doc file. Getting these documents hard copied was just a matter of two or three clicks. Eventually the software market began to flood with software of high quality and advanced features. However, switching between software was not simple. A document created on one software platform showed unnecessary changes when accessed through a different one. Saving of documents into PDF (Printable Document Format) thus became a common practice. PDFs with international scripts were easily editable, But it was difficult to do so with the ones consisting of local scripts.

In this research paper, a generic solution for the aforementioned problem is presented. Creation of generic PDF to text convertor which would be script-independent and possessing machine learning feature is the primary goal of this project.

2. LITERATURE SURVEY
Literature survey carried out shows that researches are enduring in this sector from the origin of digital documents. Before we understand this phenomenon in detail, let's understand what actually PDF and text files are:

2.1 PDF Files
PDF is abbreviated form for Printable Document Format. PDF files have .pdf extension. This format for representing documents was developed by Adobe Systems with an intention to create print-ready documents which are independent from the hardware and software which was employed for creation of the document. Each PDF document sheaths the elucidation of the underlying text document. PDF files may consist of fonts, images or vector graphics. Adobe Systems have provided Adobe Reader as a tool for viewing PDF files.

2.2 Text Files
Text files come with a .txt extension. Commonly these files contain fonts. i.e. plain text. It is one of the oldest file formats with its origins traceable from invention of computers. This file format has gained lot of popularity because of its manipulation capability by any software tool capable of handling text. A prevailing text file consists of characters and symbols of a particular character set. For instance consider the ASCII character set. MS Word, WordPad, Notepad are the software tools associated with text files.

2.3 PDF to Text Conversion-The Traditional Way
Till date various PDF to text converters have been developed. Some of these can work offline while some others work online and others in a dual way. The scope of these softwares is limited to certain scripts which are widely used and internationally accepted. Some of the major PDF to Text Converters are –

- Adobe Acrobat Reader
- Power PDF Professional

- Nitro Pro
- Corel PDF Fusion

All of them exhibit an average efficiency of approximately 90 to 99 percent. They work in offline mode i.e. internet is not required for format conversion. Some softwares work in online mode. In this scenario, the concerned PDF is uploaded on the internet. The procedure of format conversion takes place online and correspondingly editable .doc file is received as output.

In both the cases, the scope of the software is found limited to only certain scripts, as mentioned earlier. These script files are programmed in the software beforehand. Hence no foreign scripts are entertained.

3. LITERATURE SURVEY REVIEWED

Extraction of characters from an image has always been a hard nut to crack. Scripts of the same language vary from each other in many constraints like no of regions, curls, etc. For instance, alphabet A in arial font will have only two regions while in harlow script A has 3 regions. Thus for sorting out text from PDF the concept of classifier is used. In simple words classifier can be said to be a function which sorts the characters based on some constraints. Some of the classifiers found during the course of our survey are as follows:

3.1 Naive Bayes Classifier

[6]Naïve Bayes classifiers are a group of classifiers which are obtained by applying Bayes' Theorem. It gives strong assumptions based on study of features. This has been used in studies related to machine learning since 1950s. It is based on the assumptions that value of a particular feature is not related to presence or absence of any other feature, given class variable. It is further classified into Gaussian Naïve Bayes classifier, Bernoulli's Naïve Bayes classifier, Multinomial Naïve Bayes classifier, etc. based on the environment of application.

[8]Bayesian classifiers have found to be giving good results. Based on various researches conducted, their efficiency varies between 41 to 74%.

3.2 Instance based Classifier

[6]Instance Based classifier works in somewhat different way compared to the former one. It uses the principle of instance based learning. According to this principle, hypothesis is constructed directly from the learning set, instead of generating new set of values for every instance. It compares the newly arrived situations with the ones it came across during its training tenure. Instance based classifier finds applications in k nearest neighbor algorithm, kernel machines and RBF networks.

3.3 Decision Tree Learning

[6]Decision Trees have two concepts i.e. leaf nodes and intermediate nodes. The leaf nodes are final predicted answers while the intermediate nodes are questions which are obtained from the answers of their ancestors.

[8]Researches show that Decision Tree Learning provides an acceptable 76% efficiency.

Along with above phenomena there are several other concepts which were encountered during the course of literature survey.

3.4 Studies from Research Papers

Some Research papers referred by us are as follows:

3.4.1 "Automatic Alphabet Recognition"

*[1]*As the title is self-explanatory, the paper emphasized on Alphabet recognition techniques. The approach highlighted is template based approach. In this approach, universal templates are precomputed for different scripts and then comparison is made with input characters. For computation of these templates, following methods were proposed:

3.4.1.1 Position Vector Computation

In this method, Number of occurrences of each alphabet in a word is calculated. Based on above information, position of alphabet is determined.

3.4.1.2 Environment Vector Computation

This method revolves around the concept of nearest neighbor computation. For every alphabet a set of probably closest neighbors is approximated. This forms the basis for alphabet identification. For instance, the alphabet 'q' is never appears after 'u' in English script but it does in Hebrew.

3.4.2 "Character Recognition using Machine Learning Techniques"

*[2]*Basic idea of this research paper is classification of characters of the same language but different font. For instance, the character set of Arial font of English language differs from Copperplate font of the same language. To overcome this problem, the following solution is presented. Initially, input image is overlaid on known image. Pixel to pixel matching is carried out and if match is found, alphabet is said to be 'identified'. Each instance of successful pixel mapping will lead to some fixed set of values for every character. Filters can be designed for every character which will demonstrate maximum probability of an alphabet's occurrence.

3.4.3 "Character Recognition using Holland Style Adaptive Classifiers"

*[3]*This research paper reflects a simple idea of separating characters from image i.e.PDF. This is done by following the 'slicing' procedure. In this the input PDF is first horizontally sliced. After slicing, rows of text are obtained which contain characters. Correspondingly those with blank spaces are eliminated. After this, vertical slicing is applied on the rows containing characters. Thus a rectangular frame containing a single character is obtained. The total number of 'ON' pixels is computed and on the basis of this, the character is classified into right heavy or left heavy which helps in prediction of the character.

3.4.4 "Machine Learning Methods for Optical Character Recognition"

*[4]*This paper throws light on different machine learning techniques. Some important points covered include:

- Conversion of colorful image to grayscale image.
- Conversion of grayscale images to binary so as to make processing simpler.
- Application of threshold value concept for converting grayscale to binary.

Handling of colored image is quite arduous when image processing is concerned. Hence fetching values of a pixel from colored to grayscale tranquilizes the problem.

Furthermore, grayscale consists of values for every pixel in a specific range. To make processing more effortless, grayscale image is converted to binary i.e. every pixel will have either '1' or '0' value. This is achieved using the 'threshold value concept'. A certain value is maintained as threshold value. All grayscale values above the threshold value are regarded as '1' in binary image and those below it are regarded as '0'.

3.4.5 "Feature Extraction Methods for Character Recognition".

[5]As per the method of zoning mentioned in this research paper, an image converted to binary from grayscale is processed further for character identification using superimposition of n x m grid. For each of the n x m zones, average grey level is computed. This gives a feature vector of length n x m. These features are not illumination invariant. Further processing for character identification can be carried out on the basis of the information provided by the feature vector.

3.5 Inferences from Literature Survey

After conducting the literature survey, following inferences were drawn:

Conventional softwares developed have their scope limited to certain scripts. PDF documents consisting of local scripts are difficult to edit using these softwares.

Compared to Naïve Bayes classifier and Instance Based classifiers, Decision tree learning is much efficient. Superimposition of grid on a frame containing character may be difficult in some cases where characters are cluttered or have mesh like structures. In such cases, the grid method would be less preferred.

Vector computation is not much preferred way because situations differ according to the script of the language which the PDF contains. For unknown scripts determining of position using this method is rather complex. For simplifying processing, conversion of colorful image to grayscale and then to binary is fruitful. The threshold value concept is one of the efficient ways to perform this operation. Character separation from the text for processing can be done by horizontal and vertical slicing. Thus characters enclosed in frames are obtained. The 'Decision Tree Learning' concept can be used with a different approach i.e. by applying different classifiers on a particular character and then segregate those classifiers which are not beneficial in providing an expected result.

4. METHODOLOGIES

In this section the methodologies and the mathematical model that forms the core part of this project are presented. The way in which these concepts are obliging to the project and how they are implemented are described.

Some PDF documents were selected randomly and steps were carried out to edit them using conventional software. These scripts included documents consisting of scripts like English, Marathi, Urdu, Telugu, etc. Editing process consisted of selecting the text, copying the contents, inserting blank spaces, etc.

It was found that editing of English language scripted PDF documents were easily edited by the softwares. However, the software couldn't process locally scripted PDF documents as the software neither did possess learning features nor were they script-independent. This provoked the idea of creating a generic software which would process PDF documents flawlessly independent of the script of the document. But due to huge diversity among the scripts and fonts across the world, creating templates and matching patterns for each and every script is tedious. Hence, addition of machine learning feature was put forth. This would facilitate learning the unique features of every character from the provided learning character set. As a result, characters can be distinguished from the image and this would result in demonstration of machine learning feature.

5. MATHEMATICAL MODEL

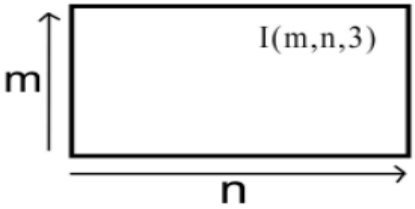

Fig. 1: Image in RGB color format

Fig.1. represents an image I(m,n,3).

Where, m=no of rows

n= no of columns

Constant 3 represents RGB color format.

I(m,n,1)=Represents a black and white image i.e. gray scale.

All calculations are performed considering image as a 2-D matrix.

5.1 Conversion of Colored Image to Gray Scale:

Conversion of colored image to gray scale image is as follows.

¥ x, y, G(x, y)=(I(x,y,0)+I(x,y,1)+I(x,y,2))/3

Where G(x, y) = gray scale of given image

x and y = co-ordinates of gray scale.

5.2 Conversion of Gray Scale to Binary:

After converting an image into gray scale, conversion of gray scale to binary format is as follows:

¥ x, y, B(x, y)=255| G(x, y)>T

0| Otherwise

Where T = threshold value (consider T =128).

B(x, y) = Binary image.

X and y = co-ordinates of binary image.

After conversion to binary image, the following procedure is applied to identify the character from the image.

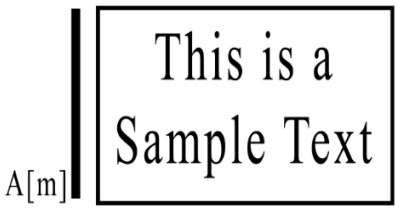

Fig. 2: Storage of character count in array

Consider an array A[m], where m is the no of rows. In A[m] there is a count of characters in particular row. Now, to obtain number of characters in each row, i.e. array indices, we derive the following formula:

$$¥ x \quad A[x] = \frac{\sum_{i=0}^{n} B(x,i)}{255} \qquad (1)$$

In above stated formula (1), the summation from 0 to n is addition of pixel values in each row divided by 255 which is the maximum value for pixel in binary image.

5.3 Slicing of PDF

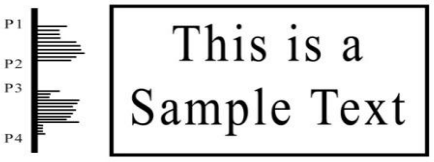

Fig. 3: Row wise slicing of PDF

Fig.3 shows the frequency of occurrence of alphabets in each row. Where, portion between P1 and P2 is a set of rows containing characters.

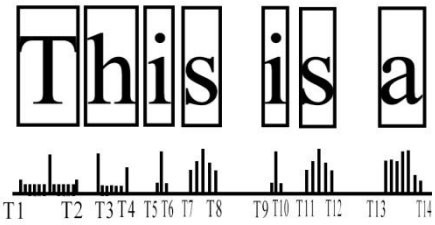

Fig. 4: Column wise slicing of PDF

Fig.4 demonstrates separation of rows from image .Now to separate character from row slices, Consider another array B[n], where n is no of columns. Portion between T1 and T2 contains a character 'T'. Similarly characters lie in between T3 and T4, T5 and T6 and so on.

5.4 Typical Character Recognition:

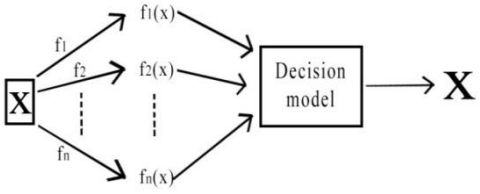

Fig. 5: Inputs to decision Model

Fig.5 shows typical character recognition contains image x on which there are some functions F1, F2, F3……Fn to be applied. The output of these functions will be stored in F1(x), F2(x)……Fn(x) respectively. Output of all functions will be combined in decision model and it will identify the character which was given in the image according to outputs of the used function.

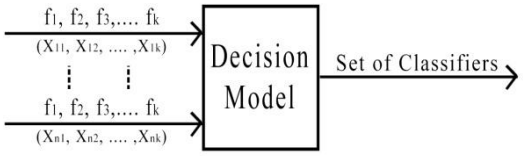

Fig. 6 Typical decision model

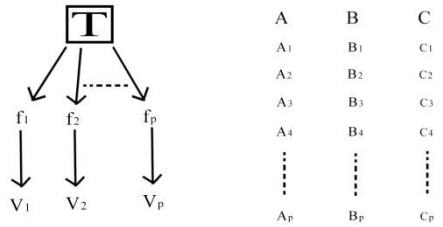

Fig. 7: Decision making process

In this decision-making process, similar functions are applied on all characters in the input image to recognize values.

Consider 'T' be the character which is given in image, then functions F1,F2,…….Fp will be applied on the given character and values will be stored in v1,v2,v3,…..vp.

A1, A2, A3, B1, B2, B3 are the standard values of A and B which are defined in some folder.

To get the exact character, Cartesian product is applied linearly between standard values of characters and current values of character which is provided in the image.

Result of Cartesian product will be compared with standard value of characters in character set and matched value will be declared as result.

6. FUTURE SCOPE

Future works in this area will include editing of handwritten, scanned documents. The way of writing a particular character differs from human to human. Thus, handling of handwritten content and correspondingly its editing would form a major part in higher versions of this project. Developing a project with machine learning feature on the grounds of light intensity and characters with variable size and shape would also be included in future works of this project. Light intensity is a major concept which cannot be neglected when scanning of documents for the purpose of processing comes into picture. This would facilitate use of this project by any person who wants to get rid of writing and storing handwritten documents. Achieving actual editing and modification of handwritten text and its further conversion into an editable document will also be included in future works of this project.

7. CONCLUSION

Study of various research papers, development of mathematical model and conducting a well-organized literature survey, following conclusions were drawn:

Storing of documents in PDF format releases the document from being changed due to change in software platform. Local script PDF documents are not easily editable by the conventional software. This can be made possible only if the converting software is platform independent. Machine learning is used to achieve this. A learning character set is provided as input which gives precise information regarding

characters in the PDF. The PDF is then sliced first horizontally and then vertically to remove blank areas in the PDF document. This result in obtaining characters enclosed in rectangular frames. Classifiers are applied to each character. Classifier sets may differ for straight line fonts, curly fonts, etc. As mentioned in literature survey, decision trees provide higher efficiency compared to others. Hence, to reach to an accurate solution, classifiers are sorted in decision tree form, with deciphered character at the root node. If character is precisely deciphered then it is processed for editing. On the other hand, if a strong prediction is not obtained for a match with any character then advanced classifiers are applied. This is a decision making process and hence, it is termed as decision model. Finally, all the identified characters are integrated together and an editable document is generated as an output.

8. REFERENCES

[1] Maayan Geffet and Yair Wiseman and Dror Feitelson, "Automatic Alphabet Recognition" School of Computer Science and Engineering, Hebrew UniversityJerusalem, Israel

[2] Jeremy Kindseth, Matthew Peterson, Muktesh Khole and Aseem Gogte"Character Recognition Using Machine Learning Techniques"

[3] Peter W. Frey and David J. Slate,"Letter Recognition Using Holland style Adaptive Classifiers" Department of Psychology, Northwestern University, Evanston, IL 60208

[4] Ivan Dervisevic"Machine Learning Methods for Optical Character Recognition".

[5] Oivind Due Trier, Anil K. Jain and Torfinn Taxt, "Feature Extraction Methods for Character Recognition""Comparison of Machine Learning Classifiers for Recognition of Online and Offline Handwritten digits"Computer Engineering and Intelligent Systems www.iiste.orgISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)Vol.4, No.13, 2013

[6] Kirill Safronov, Dr.-Ing. Igor Tchouchenkov, Prof. Dr.-Ing. Heinz Wörn"Optical Character Recognition Using Optimisation Algorithms"Institute for Process Control and Robotics (IPR)University of Karlsruhe, Karlsruhe, Germany

[7] S. M. Kamruzzaman, "Text Classification using Artificial Intelligence"Department of Information and Communication EngineeringUniversity of Rajshahi, Rajshahi-6205, Bangladesh.