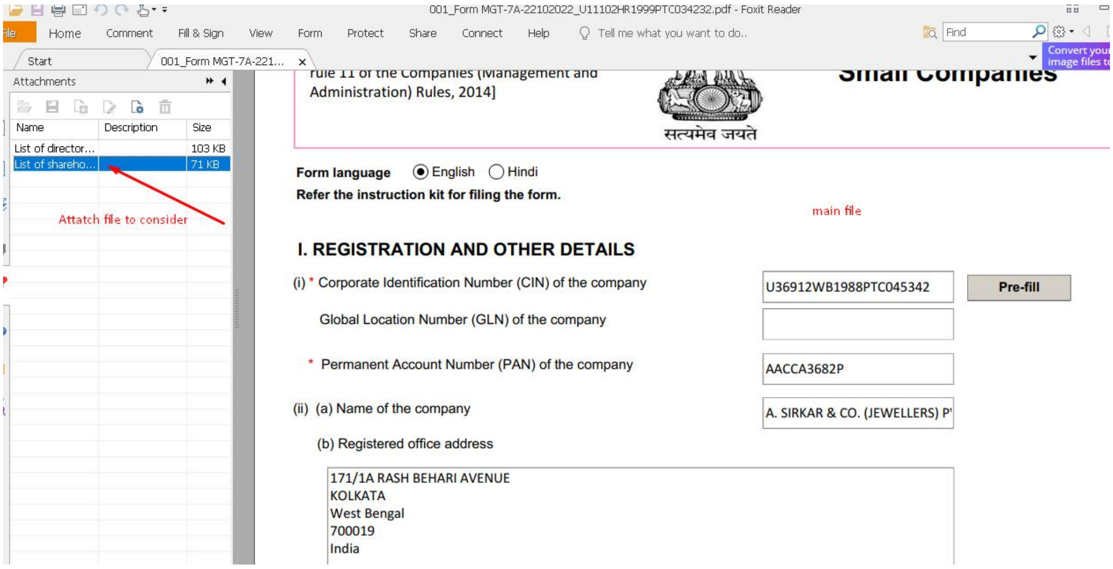
What does this program do?

This program extracts some information in main pdf file and attatch files of main file, and then convert it into xlsx file.

Template of Image files:



Name	Description	Size
List of director...		103 KB
List of shareho...		71 KB

rule 11 of the Companies (Management and Administration) Rules, 2014]

Form language ● English ○ Hindi

Refer the instruction kit for filing the form.

I. REGISTRATION AND OTHER DETAILS

(i) * Corporate Identification Number (CIN) of the company: U36912WB1988PTC045342

Global Location Number (GLN) of the company:

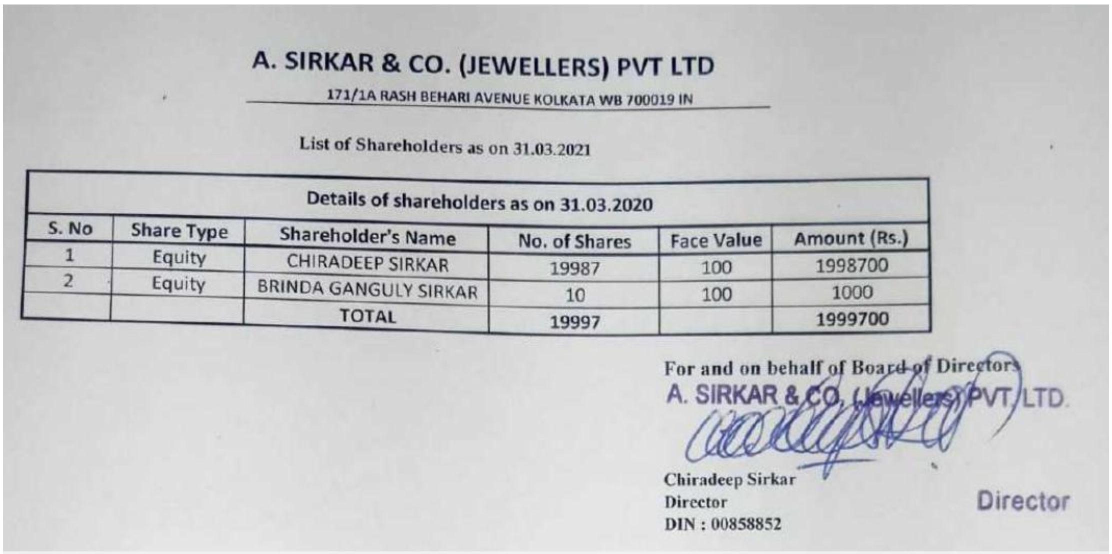
* Permanent Account Number (PAN) of the company: AACCA3682P

(ii) (a) Name of the company: A. SIRKAR & CO. (JEWELLERS) P

(b) Registered office address

171/1A RASH BEHARI AVENUE
KOLKATA
West Bengal
700019
India

Fig.1 main pdf file



A. SIRKAR & CO. (JEWELLERS) PVT LTD

171/1A RASH BEHARI AVENUE KOLKATA WB 700019 IN

List of Shareholders as on 31.03.2021

Details of shareholders as on 31.03.2020

S. No	Share Type	Shareholder's Name	No. of Shares	Face Value	Amount (Rs.)
1	Equity	CHIRADEEP SIRKAR	19987	100	1998700
2	Equity	BRINDA GANGULY SIRKAR	10	100	1000
		TOTAL	19997		1999700

For and on behalf of Board of Directors
A. SIRKAR & CO. (Jewellers) PVT. LTD.

Chiradeep Sirkar
Director
DIN : 00858853

Director

Fig 2. Attach file

Quality of files

- Main pdf file is digital pdf and its quality is good.
- Quality of attach file is very various and poor.

Requirements

1) Main pdf file

- Extraction of CIN number, shareholding as at(date)_History_pc, Total shaes(No.of shaes of subscrbed capital)

2) Attach pdf file

- Extraction of Shareholder name, Shareholder address, No of shares Held_history_ik

What is important in this program?

What engine does this program use ?

Opencv, pymupdf, cnn, r-cnn, easyocr, tesseract

- Opencv:

This library take charge of image pre-processing

- CNN and RCNN:

For table extraction

- easyocr and tesseract:

Text recognition

How does this program work?

Algorithm of this program

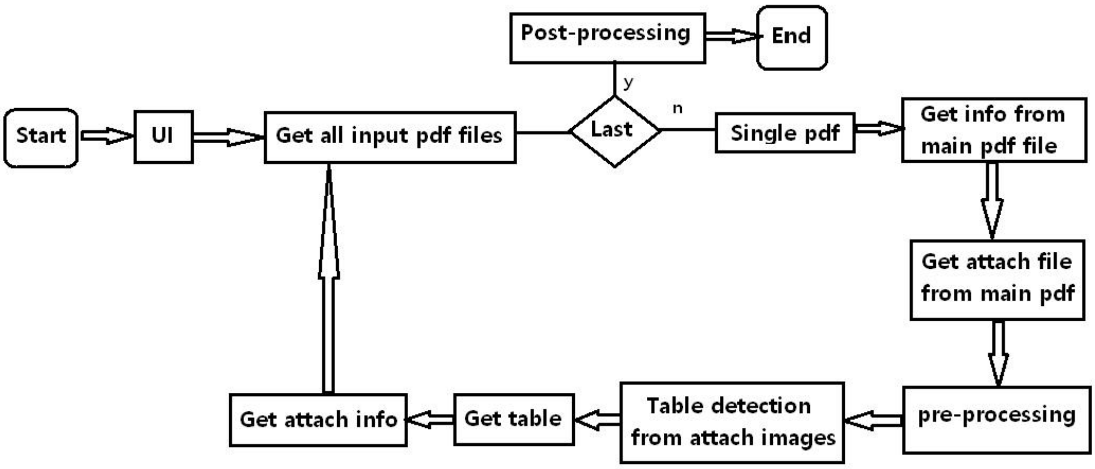
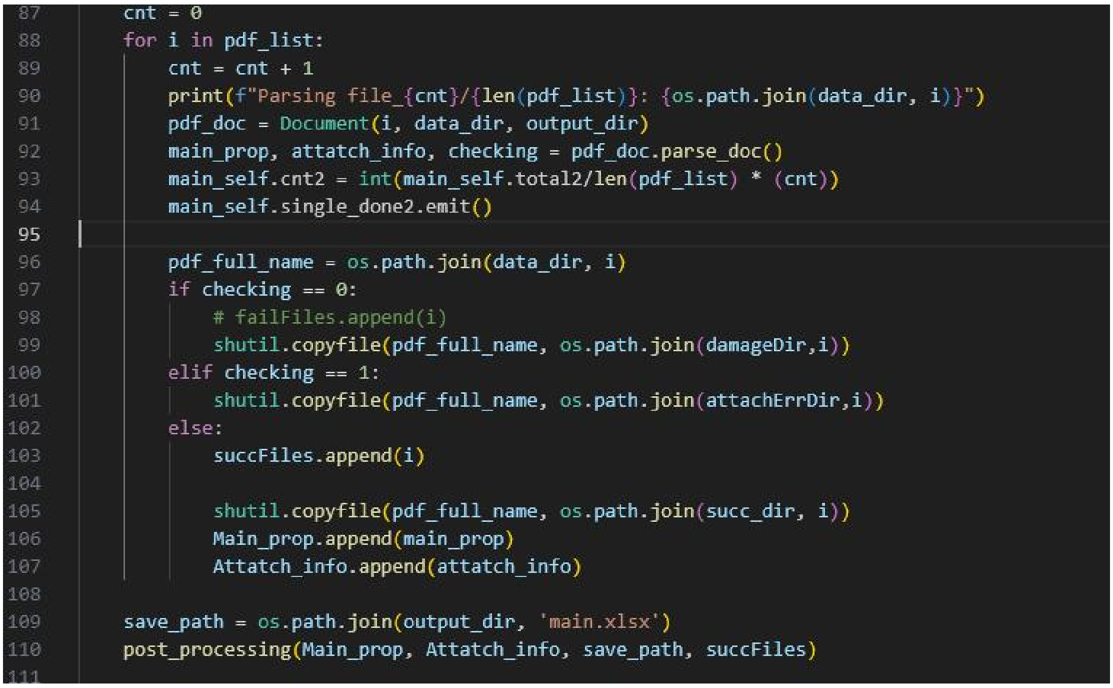


Fig. 3 Overall Algorithm

In code,

MGDATA.py: take charge of parts of program starting, UI and several pre-processing such as getting input PDF files.

main_MG.py: reflect logics of overall algorithm.



```
    cnt = 0
    for i in pdf_list:
        cnt = cnt + 1
        print(f"Parsing file_{cnt}/{len(pdf_list)}: {os.path.join(data_dir, i)}")
        pdf_doc = Document(i, data_dir, output_dir)
        main_prop, attatch_info, checking = pdf_doc.parse_doc()
        main_self.cnt2 = int(main_self.total2/len(pdf_list) * (cnt))
        main_self.single_done2.emit()

        pdf_full_name = os.path.join(data_dir, i)
        if checking == 0:
            # failFiles.append(i)
            shutil.copyfile(pdf_full_name, os.path.join(damageDir,i))
        elif checking == 1:
            shutil.copyfile(pdf_full_name, os.path.join(attachErrDir,i))
        else:
            succFiles.append(i)

            shutil.copyfile(pdf_full_name, os.path.join(succ_dir, i))
            Main_prop.append(main_prop)
            Attatch_info.append(attatch_info)

    save_path = os.path.join(output_dir, 'main.xlsx')
    post_processing(Main_prop, Attatch_info, save_path, succFiles)
```

Fig 4. Algorithm (explanation in code)

Get info from main pdf (line 841 in MG_parser.py)

Function- mainProp():

The goal of the mainProp function is to extract specific information from a PDF file using the pdf-xfa-tools library and return it as a list. The function performs the following steps:

- Opens the PDF file specified by self.doc_dir and self.doc_name.
- Retrieves the XFA (XML Forms Architecture) data from the PDF.
- Parses the XFA data using BeautifulSoup.
- Searches for the value of the CIN (Company Identification Number) within the XML data. It looks for a <cin> tag and checks if the length of its string value is greater than 10 characters (assuming CIN length is 21). If found, assigns the value to the CIN_no variable and breaks out of the loop.
- Searches for the value of the date of AGM (Annual General Meeting) within the XML data. It looks for a <date_agm> tag and checks if the length of its string value is greater than 5 characters. If found, assigns the value to the share_his_pc variable and breaks out of the loop.
- Searches for the value of the total number of shares within the XML data. It looks for a <tot_no_es_s_cap> tag and checks if the length of its string value is greater than 5 characters. If found, assigns the value (after splitting by '.') to the total_share variable, considering only the whole number part, and breaks out of the loop.
- Returns a list containing the values of CIN_no, share_his_pc, and total_share.

Overall, the function aims to extract specific information related to the company's identification number (CIN), the date of the Annual General Meeting (AGM), and the total number of shares from the provided PDF file.

Get info from main pdf (line 841 in MG_parser.py)

attachProcessing() function

It returns name and contect of attach file to consider.

Table detection from attach image

Function cascade_mmdet():

The goal of the function cascade_mmdet is to perform object detection on an input image using the Cascade Mask R-CNN model, and specifically to detect tables and cells within the image. It takes an image as input and returns a list of detected tables.

The structure of the function is as follows:

- It initializes a scaling factor, which is set to 1.5 in this case.
- The input image is copied to a variable named img.
- The inference_detector function is called to obtain the detection results using the pre-trained model and the input image img. The results contain bounding box coordinates and confidence scores for different classes of objects.
- Three empty lists are initialized to store the detected objects: res_borderTable for tables with borders, res_borderlessTable for borderless tables, and res_cell for cells.
- A loop iterates over the detected objects from the "border table" class (obtained from the detection results) and appends the bounding box coordinates to res_borderTable if the confidence score is above a threshold of 0.55.
- Another loop iterates over the detected objects from the "cell" class and appends the bounding box coordinates to res_cell if the confidence score is above a threshold of 0.85.
- A third loop iterates over the detected objects from the "borderless table" class and appends the bounding box coordinates to res_borderlessTable if the confidence score is above a threshold of 0.55.
- The function then draws rectangles around the detected tables on the img using the coordinates from res_borderTable and res_borderlessTable, visualizing them in red color with a thickness of 3 pixels.
- Finally, the function returns the list of detected tables.

Get table

Function- getting_table():

The goal of the getting_table function is to process an input image and extract a binary image representation of the table structure. The function performs the following steps:

- Remove unnecessary columns: Unwanted columns are removed from the image.
- Get all horizontal lines: Horizontal lines present in the image are identified.
- Get the upper and lower limits of the table and determine the number of rows.

- Create a table image using the column and row information.

The function handles both cases when the image contains borders around the table and when it doesn't. If there are no borders, the function performs additional operations to identify and remove lines that could interfere with the table extraction process. On the other hand, if the image contains borders, the function extracts the column and row coordinates, generates vertical and horizontal lines based on these coordinates, and combines them to create the table image.

The function returns the binary table image (img_vh) and the original processed image (img).

Get attach info

Function- Box_detection()

The goal of the box_detection function is to identify and detect boxes within an image. The function takes an input image in the variable img_vh and performs the following steps:

- Retrieves the height and width of the input image.
- Creates a copy of the original image.
- Finds contours in the binary image img_vh.
- Iterates over each contour and checks if its width (w) is between 15 and 80% of the image width, and if its height (h) is greater than 15. If these conditions are met, it draws a rectangle around the contour on the copied image.
- Stores the coordinates (x, y, w, h) of the detected boxes in a list called box.
- Sorts the box list based on the y-coordinate (y) of each box.
- Converts the box list into a numpy array and reshapes it into a 2D array, where each row represents a unique y-coordinate and each column represents a box's coordinates (y, x, h, w).

Finally, the function returns the 2D array of detected boxes.

In summary, the goal of this function is to identify and extract the bounding boxes of objects or regions of interest in an image.

Funcion- GetExtractBoxes()

The goal of the function getExactBoxes is to process a set of boxes, along with other parameters, and return modified versions of the boxes and related indices

Function- Box_text_detection()

The goal of the box_text_detection function is to perform text detection within bounding boxes specified by boxes.

Function- InfoFromText()

The goal of the function infoFromText is to extract specific information from a given text based on the provided indices (inds) and perform some data processing tasks. The extracted information is then stored in the attatch_info list and returned as the output.

Here are the main steps of the function:

- Initialize an empty attatch_info list with three sublists.
- Check if the first index (inds[0]) is not equal to -1 and if the length of the first row of text is greater than 0.
- Remove any empty rows from the text array.
- Delete the first row of text.
- If the type of inds[0] is not a list: a. Convert text to a nested list. b. Iterate over each row of text and group consecutive non-empty values together, creating a new nested list (newText). c. Convert newText back to a NumPy array and assign it to text.
- Iterate over the indices in inds. a. If the current index is not equal to -1:

  If i is 0 and nameAddr is true, extract the name and address information by splitting the values at newline characters and store them in the first and second sublist of attatch_info, respectively.

  If i is 0 but nameAddr is false, attempt to concatenate non-empty values within each row of the current index column (ind) using underscore characters and store the result in the first sublist of attatch_info.

  If i is 2, split each value in the current index column (ind) at space characters and store only the first part in the third sublist of attatch_info.

  For other values of i, store the corresponding column (ind) of text in the attatch_info list.
- Return the attatch_info list containing the extracted information.

Post-processing

- This part makes xlsx file and json of output.