ISSN 2278-3091
Volume 9 No.2, March -April 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse18922020.pdf
https://doi.org/10.30534/ijatcse/2020/18922020

AI approach with increased accuracy to extract the tabular content from PDF and Image files

Shriram K Vasudevan[1], Vamsee Krishna Kiran M[2], Sini Raj P[3] ,Thangavelu S[4]
[1,2,3,4] Dept. of Computer Science and Engineering
Amrita School of Engineering
Coimbatore, Amrita Vishwa Vidyapeetham, India.
{kv_shriram, mk_vamseekrishna, p_siniraj,s_thangavel}@cb.amrita.edu

ABSTRACT

In today's era of computerized banking, management, billings and what not, we use tabular data in every sector. The most commonly used format of storing tabular data by us is through excel format. It is very easy to retrieve information from excel sheets. But, tabular data extraction from PDFs or images has remained as an inherent problem since many years. In order to reduce such issues and automate the process we have designed a system using artificial intelligence that can take a PDF or an image as an input and outputs a CSV or excel file directly with the extracted tabular data.

Key words: PDF Table data extraction, Artificial Intelligence, Computerized banking, automated process.

1. INTRODUCTION

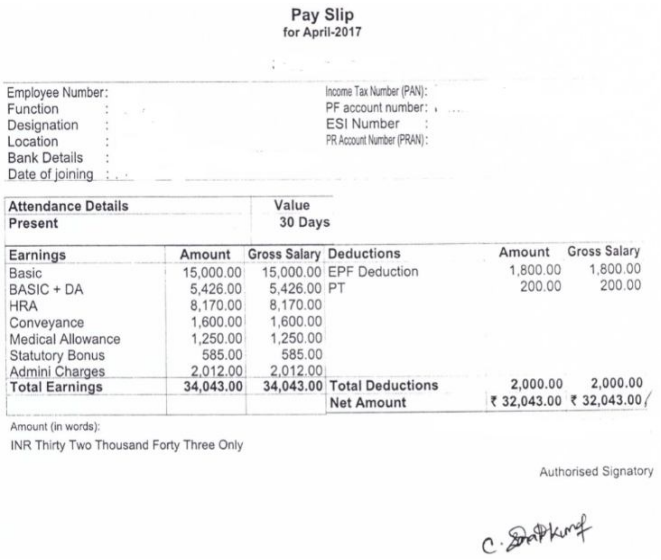
Every one of us come across tabular data at least one or more times a day. It is easier to extract tabular data from excel or CSV formats but it is very difficult to extract data from a PDF or an image. The only way one would follow to achieve this task is to open a new excel sheet and creating a table, enter the values manually which is a very hectic task for an individual to do.

Rank	Name of the country	Sex ratio at birth
No.1	Liechtenstein	126 males/100 females
No.2	China	115 males/100 female
No.3	Armenia	113 males/100 females
No.4	India	112 males/100 females
No.5	Azerbaijan	111 males/100 females
No.5	Viet Nam	111 males/100 females
No.6	Albania	110 males/ 100 females
No.7	Georgia	108 males/100 females
No.8	South Korea	107 males/100 females
No.8	Tunisia	107 males/ 100 females
No.9	Nigeria	106 males/ 100 female
No.10	Pakistan	105 males/100 females
No.11	Nepal	104 males/ 100 females

Figure 1: Sample tabular data found in PDF's

Figure 1 shows a sample tabular data of Male to Female ratios of different countries across the world. To extract this type of data, it is a very tedious process and consumes lots of manpower, time which are very crucial in this competitive world. Figure 2 gives much more real time intuition about the discussed problem.

Figure 2 is another example of a real-time tabular data when captured with mobile phone cameras. Besides the hassles that are caused by the above mentioned tabular data there lies another major problem. This type of data must be entered manually which includes human intervention due to which many errors creep into the entries. So, this needs several cross-checking to be done before producing the final copy. One cannot invest time and money for such inefficient work. In order to overcome all these errors and problems, we have developed an automated table detection system using deep learning techniques s, which can extract tabular data from a PDF or an image.



Pay Slip
for April-2017

Employee Number:
Function :
Designation :
Location :
Bank Details :
Date of joining :

Income Tax Number (PAN):
PF account number:
ESI Number :
PR Account Number (PRAN):

Attendance Details	Value
Present	30 Days

Earnings	Amount	Gross Salary	Deductions	Amount	Gross Salary
Basic	15,000.00	15,000.00	EPF Deduction	1,800.00	1,800.00
BASIC + DA	5,426.00	5,426.00	PT	200.00	200.00
HRA	8,170.00	8,170.00			
Conveyance	1,600.00	1,600.00			
Medical Allowance	1,250.00	1,250.00			
Statutory Bonus	585.00	585.00			
Admini Charges	2,012.00	2,012.00			
Total Earnings	34,043.00	34,043.00	Total Deductions	2,000.00	2,000.00
			Net Amount	₹ 32,043.00	₹ 32,043.00

Amount (in words):
INR Thirty Two Thousand Forty Three Only

Authorised Signatory

Figure 2: Example of a real time tabular image

2. EXISTING SOLUTIONS

Here are some of the different approaches, existing solutions for the table detection from the PDF or image files with some disadvantages in implementing them (Ref. Table.1).

Table 1: A comparison of existing techniques

	Uses Pipelining	Uses OCR	For Both PDF's and Images	Skew Correction
Gantos B, Danatsas	✕	✓	✓	✕
Wenzel C	✕	✓	✕	✕
Proposed System	✓	✓	✓	✓

The model proposed by Gatos, B., Danatsas, D., consists of an algorithm that can process only tabular data present in the documents, PDF files and handwritten scripts. Handwritten documents can be easily processed through available OCR tools like tesseract etc., Although it is a great approach with the binary pattern recognition and pre-processing, it is mostly not suitable with datasets containing images. [1]

Wenzel, C. has proposed a reference tables based table detection algorithm. It cannot be declared that it is an effective method however there are some disadvantages with these algorithms. Every table that will be an input to the algorithm need not to be in the same form of the reference table and even though a CNN is trained with lots of tabular data, the inaccuracy of detecting the table is very high. Whereas our algorithmic approach to the tabular detection is even more accurate and reduces the execution time. [2]There are various other techniques where we could use SVM [12], spreadsheet DB [11] and semantic based [10] content retrieval approaches mentioned in various literatures.

3. PROBLEM STATEMENT

To design an effective and optimized algorithm that can retrieve information of tabular data from PDF and image files.

4. ARCHITECTURE

In order to handle the data effectively, two distinct algorithms have been designed to process PDF files and Images. PDF files have a different architecture to that of Image files, the execution starts by identifying the extension of the input file.

4.1 Image File

Images are the most convenient and effective source for generating the tabular data. In order to process these image files, we have followed the following steps.

A. Skew Correction

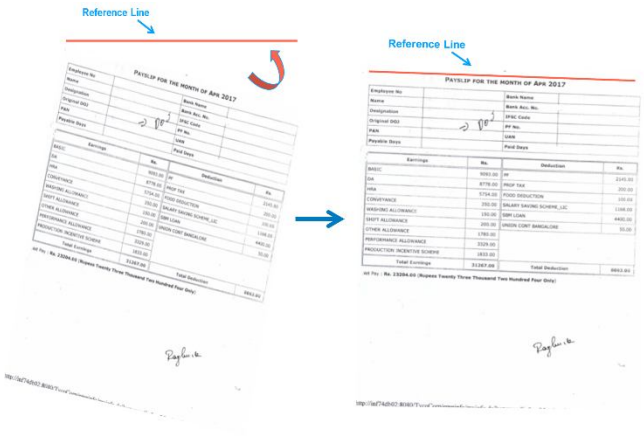


Figure 3: Skew correction on an image with tabular data

Data present in any format can easily be converted to image just by taking a picture with a mobile. But, there is a major issue in the process, any image that we take cannot be exactly aligned and most of the times the image is a bit tilted. It is hard to process these kind of files due to lack of skew in the image. In order to overcome this problem, the first step the algorithm does is to correct the skew of the image. This can be achieved by setting a reference horizontal line and rotating the image until it reaches the horizontal line. Figure 3 depicts the process of skew correction. [3]

B. Searchable PDF generation

Once we have corrected the skew, we need to generate a searchable PDF version of image file. It is needed because PDF files have a feature that, if we type a string in the find box PDF's will be able to search and display the matching strings available whereas we cannot perform search operations over an image file. In order to convert the image to searchable PDF we have used tesseract OCR tool which is basically an optical character recognition tool for image processing. [4]

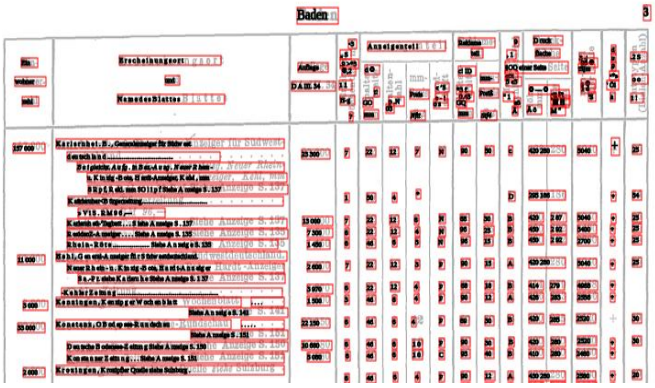


Figure 4: Creation of a searchable PDF over an Image file

C. Detection of columns and rows:

Apparently this is the most prominent part of table detection process. Hough transform technique is used for detecting the rows and columns in the searchable PDF. Hough transform traces the pixels of a straight line in a given image by which it detects the rows and columns.

In order to detect a straight line, it searches for the pixels that look like a straight line. It starts the search at a random angle and gradually reaches the point where the straight line is present. Once all the straight lines are detected it looks for the intersection of these lines, which means that it is eventually looking for the vertices of all the boxes present in the table. Figure 5 gives an intuition on how the algorithm creates the Hough transform based straight line contouring in the table. [5]

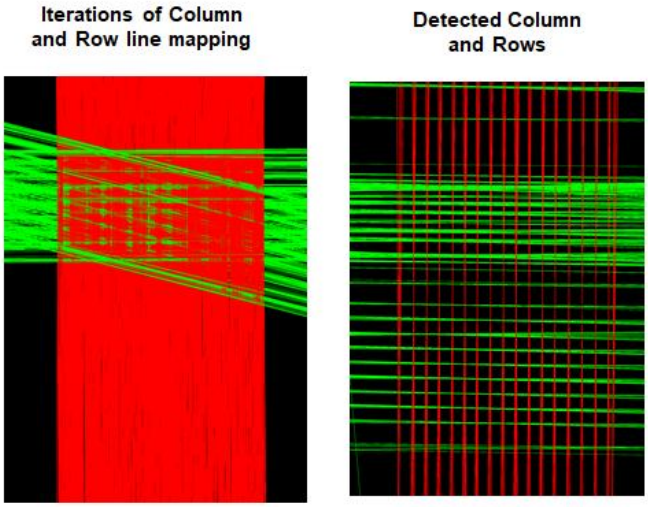


Figure 5: Column,Row detection using hough

D. Mapping the lines and Extracting the Text

Now, the last step is to map the boxes detected by the Hough transform and impose them on the original image in order to extract the text present inside them. Figure 6 shows the imposition of Hough transform lines on the original image. Once we project these lines onto the original image, we enter the last step in detecting the tabular data. As we are left with the detected boxes where the text is present, we can run an optical character recognition algorithm on the image to extract this data. In this case, we have used tesseract OCR in order to achieve this task.

Tesseract OCR is one of the effective and fastest character recognition tool which is mostly used in image processing like applications. Figure 7 shows the output data exported to a CSV format. It can be normally accessed with software's like Microsoft excel, Libreoffice sheets etc., which are mostly used by the users.

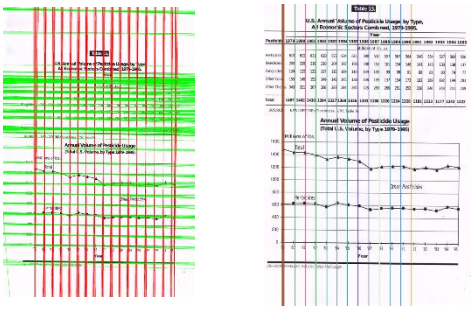


Figure 6: Projection of detected lines and vertices onto the original image

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	col01	col02	col03	col04	col05	col06	col07	col08	col09	col10	col11	col12	from_page
2	622) 631)	620)	573) 634)	611)	590)	532) 557)	/567/	564)	546) 554	527;	583)	556	1
3													1
4	228) 218)	210)	204) 197)	193)	188	152) 161)	154	148)	141) 143)	130)	138)	137	1
5													1
6	122) 122)	117)	115) 109)	110)	109)	100 99	98	91	86 81	80)	_ 79	77	1
7													1
8	149) 152)	149)	148 145)	138)	138	133) 137	154;	173;	182) 189)	192)	199)	203	1
9													1
10	321) 307)	298)	287) 284)	284)	278	260 266)	251)	252)	226/ 246)	248)	244)	249	1
11													1
12	1487) 1442)	1430)	1394/ 1327/1	1336									1
13							1303]	1186]	1220] 1224]	1228]	1181 1213	1177] 1243] 1222	1
14	EPA/OPP/BEADestimates, 3/97, Table 14.												1
15					Annual Volume of Pesticide Usage (Total U.S. Volume, by Type 1979-1995)								1
16	(Millions of lbs.)												1
17													1
18	Total												1

Figure 7: Output data visualized as a CSV file

4.2 PDF Files

PDF is the most used format for storing the data. Most of the data say from documents to books is available in PDF format. Unlike the image files PDF files does not need steps like searchable PDF generation, line Contouring, etc. We use another similar kind of architecture for detecting and retrieving the tabular data from the PDF files.

PDF is the most used format for storing the data. Most of the data say from documents to books is available in PDF format. Unlike the image files PDF files does not need steps like searchable PDF generation, line Contouring, etc. We use another similar kind of architecture for detecting and retrieving the tabular data from the PDF files.

A. Page Segmentation

Page segmentation is nothing but counting the number of pages in a given PDF and save a duplicate copy of them for processing. This is done because the tool used for table

detection cannot process multiple pages at a time. In order to make it work for multiple page PDF we are following the process of iterating over the duplicated copies of the individual pages.

B. Noise removal or Pre-processing

Most of the image processing applications are incomplete without this step. Pre-processing decreases the unwanted signals and increase the effectivity of data retrieval. Our algorithm uses dilation and erosion as the noise removal techniques. [6]

C. Table data extraction: In order to retrieve the data from the PDF page our algorithm uses tabula tool. It can only accept single page PDF files and this the reason why we have performed pre-processing step in the beginning. Optimization of the running time is also an important parameter to be taken care of. The algorithm is designed to process the data using pipelining technique, which can greatly reduce the processing time of the algorithm. [7]

D. Pipelining

It loads the first page to process and performs the pre-processing step. By the time the second step completes second page is loaded for pre-processing. By the time first page completes its data retrieval the second page completes the pre-processing step and gets ready for data retrieval by simultaneously loading the third page for processing and this process goes on as long as all the pages are processed and respective CSV files are created. The above mentioned process can be effectively understood by referring Figure 8. [8]

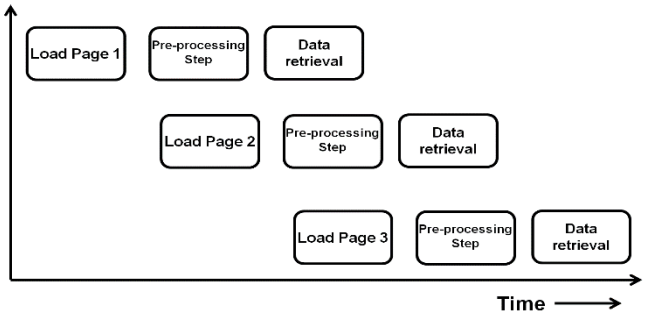


Figure 8 : Pipelining process

E. Morphological Operators

In a PDF for detecting table like structures, tabula package in python uses morphological operators. This algorithm is the best to detect the table like structures in a given PDF. These are the operators which are designed with names like North-East, North-West, West-South, East-South, etc., to detect the edges, vertices of a table. [9]

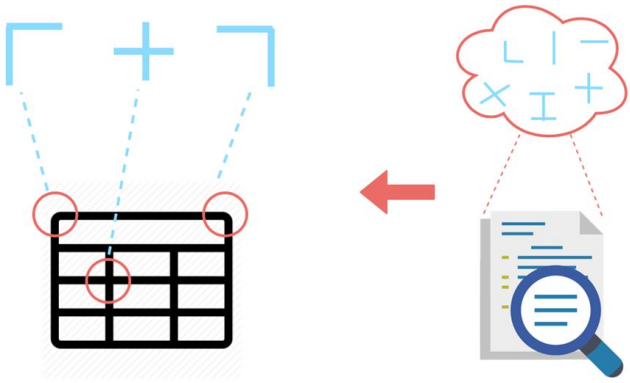
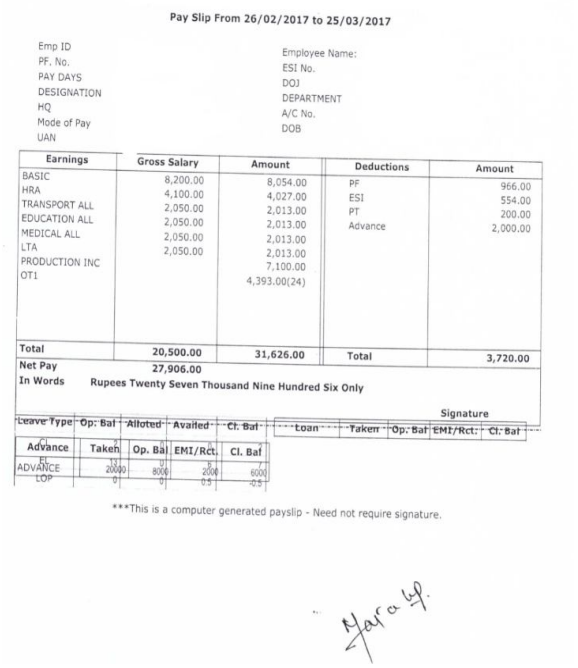


Figure 9: Morphological operators for table detection

5. RESULTS

We have tested our algorithm on different kinds of input PDF's, Image files and found out the accuracy in retrieving the tabular data in PDF's is more than 98% while in Images it is around 95%. The extracted tabular data is converted into .xlsx, .csv, .xml, .json formats.

5.2 Image Files



Pay Slip From 26/02/2017 to 25/03/2017

Emp ID
PF. No.
PAY DAYS
DESIGNATION
HQ
Mode of Pay
UAN

Employee Name:
ESI No.
DOJ
DEPARTMENT
A/C No.
DOB

Earnings	Gross Salary	Amount	Deductions	Amount
BASIC	8,200.00	8,054.00	PF	966.00
HRA	4,100.00	4,027.00	ESI	554.00
TRANSPORT ALL	2,050.00	2,013.00	PT	200.00
EDUCATION ALL	2,050.00	2,013.00	Advance	2,000.00
MEDICAL ALL	2,050.00	2,013.00		
LTA	2,050.00	2,013.00		
PRODUCTION INC		7,100.00		
OT1		4,393.00(24)		
Total	20,500.00	31,626.00	Total	3,720.00

Net Pay 27,906.00
In Words Rupees Twenty Seven Thousand Nine Hundred Six Only

Signature

Leave Type	Op. Bal	Alloted	Availed	Cl. Bal
CL	12	1	5	8
ADVANCE	20000	0	8000	12000
LOP	0	0	0.5	-0.5

Loan	Taken	Op. Bal	EMI/Rct.	Cl. Bal
Advance	8000	8000	2000	6000

***This is a computer generated payslip - Need not require signature.

Figure 10: Input Image for our Algorithm

Emp ID			Employee Name:		
PF. No.		ESI No.			
PAY DAYS					
		DOJ			
DESIGNATION					
		DEPARTMENT			
HQ		A/C No.			
Mode of Pay					
		DOB			
UAN					
Earnings	Gross Salary	Amount			
			Deductions		Amount
BASIC		8,200.00 8,054.00			
HRA		4,100.00	PF		966
		4,027.00	ESI		554
TRANSPORT ALL		2,050.00 2,013.00	PT		
EDUCATION ALL		2,050.00 2,013.00	Advance		200
MEDICAL ALL		2,050.00 2,013.00			2,000.00
		2,050.00			
		2,013.00			
PRODUCTION INC		7,100.00			
OT1		4,393.00(24)			
Total		20,500.00			
		31,626.00	Total		3,720.00
Net Pay		27,906.00			
In Words					
		teaveType| Op: Bat} â€œAltoted| Avaited=}-ehBat y Signature			Bat} EMr/Ret: |Cl: Bar4 â
Advance E Taken! 13	U0	Op. BS|EMI/R&t| Cl. Bai			
ADVANCE COP 20000 0	800! ot	05 200! 6 0 600			
		***This is a computer generated payslip - Need not r Co A.			

Figure 11: Output in .csv, .xlsx format

```
<text top="342" left="71" width="37" height="12" font="4">BASIC</text>
<text top="346" left="271" width="54" height="12" font="4">8,200.00</text>
<text top="348" left="415" width="54" height="12" font="4">8,054.00</text>
<text top="351" left="531" width="13" height="12" font="4">PF</text>
<text top="355" left="745" width="41" height="12" font="4">966.00</text>
<text top="359" left="71" width="24" height="12" font="4">HRA</text>
<text top="363" left="271" width="54" height="12" font="4">4,100.00</text>
<text top="366" left="415" width="54" height="12" font="4">4,027.00</text>
<text top="368" left="531" width="19" height="12" font="4">ESI</text>
<text top="372" left="745" width="40" height="12" font="4">554.00</text>
<text top="377" left="69" width="101" height="13" font="4">TRANSPORT ALL</text>
<text top="381" left="271" width="53" height="12" font="4">2,050.00</text>
<text top="383" left="415" width="54" height="12" font="4">2,013.00</text>
<text top="385" left="531" width="14" height="12" font="4">PT</text>
<text top="389" left="744" width="41" height="12" font="4">200.00</text>
<text top="394" left="70" width="99" height="14" font="4">EDUCATION ALL</text>
<text top="398" left="271" width="54" height="12" font="4">2,050.00</text>
<text top="401" left="415" width="54" height="12" font="4">2,013.00</text>
<text top="403" left="530" width="51" height="12" font="4">Advance</text>
<text top="407" left="732" width="53" height="12" font="4">2,000.00</text>
<text top="413" left="70" width="81" height="13" font="4">MEDICAL ALL</text>
<text top="417" left="270" width="54" height="12" font="4">2,050.00</text>
<text top="420" left="415" width="54" height="12" font="4">2,013.00</text>
<text top="430" left="69" width="21" height="12" font="4">LTA</text>
<text top="435" left="270" width="53" height="12" font="4">2,050.00</text>
<text top="438" left="415" width="54" height="12" font="4">2,013.00</text>
<text top="448" left="69" width="110" height="14" font="4">PRODUCTION INC</text>
<text top="454" left="414" width="54" height="12" font="4">7,100.00</text>
<text top="462" left="68" width="23" height="15" font="5">OT1</text>
<text top="469" left="386" width="81" height="15" font="5">4,393.00(24)</text>
<text top="556" left="66" width="34" height="12" font="4">Total</text>
<text top="560" left="254" width="68" height="12" font="4">20,500.00</text>
<text top="562" left="398" width="69" height="12" font="4">31,626.00</text>
<text top="565" left="528" width="34" height="12" font="4">Total</text>
<text top="568" left="725" width="59" height="12" font="4">3,720.00</text>
<text top="575" left="67" width="52" height="14" font="0">Net Pay</text>
<text top="580" left="253" width="68" height="14" font="0">27,906.00</text>
```

Figure 12: Output in .xml format

The results, output of extracted tabular data for tested Images can be found out at https://github.com/strangest-quark/TableExtraction/tree/master/ImageModule

5.2. PDF files

Country	The average weighted total yearly salary in terms of PPS
Austria	60.530
Belgium	55.998
Bulgaria	9.770
Croatia	27.063
Cyprus	50.549
Czech Republic	36.950
Denmark	43.669
Estonia	21.053
Finland	36.646
France	47.550
Germany	53.358
Greece	30.835
Hungary	27.692
Iceland	33.801
Ireland	49.654
Israel	59.580

Country	The average weighted total yearly salary in terms of PPS
Italy	34.120
Latvia	21.580
Lithuania	29.660
Luxembourg	56.268
Malta	40.342
Netherlands	56.721
Norway	41.813
Poland	21.591
Portugal	33.334
Romania	13.489
Slovakia	18.282
Slovenia	37.970
Spain	38.873
Sweden	47.143
Switzerland	59.902
Turkey	26.250
United Kingdom	52.776

Table 10 - The average weighted total yearly salary of researchers of each country in EU25 and Associated Countries (2006, N=6110, all currencies in PPS)

Country	Net Yearly salary average in terms of PPS
Austria	30.603
Belgium	26.336
Bulgaria	9.801
Croatia	20.254
Cyprus	39.732
Czech Republic	22.252
Denmark	24.917
Estonia	13.777
Finland	22.971
France	26.983
Germany	28.687
Greece	24.326
Hungary	16.723
Iceland	22.354
Ireland	28.193
Israel	37.389

Country	Net Yearly salary average in terms of PPS
Italy	22.372
Latvia	18.828
Lithuania	13.507
Luxembourg	40.942
Malta	28.498
Netherlands	35.573
Norway	26.088
Poland	14.104
Portugal	21.835
Romania	12.500
Slovakia	12.173
Slovenia	18.211
Spain	27.060
Sweden	22.801
Switzerland	46.432
Turkey	23.530
United Kingdom	35.372

Table 11 - Country Net Yearly Salary Averages of researchers in EU25 and Associated Countries (2006, N=6.934, all currencies in PPS)

Figure 13: Input PDF for our Algorithm

	The average weighted			The average weighted
Country	total yearly salary in		Country	total yearly salary in
	terms of PPS			terms of PPS
Austria	60.53	Italy		34.12
Belgium	55.998	Latvia		21.58
Bulgaria	9.77	Lithuania		29.66
Croatia	27.063	Luxembourg		56.268
Cyprus	50.549	Malta		40.342
Czech Republic	36.95	Netherlands		56.721
Denmark	43.669	Norway		41.813
Estonia	21.053	Poland		21.591
Finland	36.646	Portugal		33.334
France	47.55	Romania		13.489
Germany	53.358	Slovakia		18.282
Greece	30.835	Slovenia		37.97
Hungary	27.692	Spain		38.873
Iceland	33.801	Sweden		47.143
Ireland	49.654	Switzerland		59.902
Israel	59.58	Turkey		26.25
	Net Yearly salary			Net Yearly salary
Country	average in terms of		Country	average in terms of
	PPS			PPS
Austria	30.603	Italy		22.372
Belgium	26.336	Latvia		18.828
Bulgaria	9.801	Lithuania		13.507
Croatia	20.254	Luxembourg		40.942
Cyprus	39.732	Malta		28.498
Czech Republic	22.252	Netherlands		35.573
Denmark	24.917	Norway		26.088
Estonia	13.777	Poland		14.104
Finland	22.971	Portugal		21.835
France	26.983	Romania		12.5
Germany	28.687	Slovakia		12.173
Greece	24.326	Slovenia		18.211
Hungary	16.723	Spain		27.06
Iceland	22.354	Sweden		22.801
Ireland	28.193	Switzerland		46.432
Israel	37.389	Turkey		23.53

Figure 14: Output in .xlsx, .csv format

```
"",The average weighted,,,The average weighted
Country,total yearly salary in,,Country,total yearly salary in
"",terms of PPS,,,terms of PPS
Austria,60.530,Italy,,34.120
Belgium,55.998,Latvia,,21.580
Bulgaria,9.770,Lithuania,,29.660
Croatia,27.063,Luxembourg,,56.268
Cyprus,50.549,Malta,,40.342
Czech Republic,36.950,Netherlands,,56.721
Denmark,43.669,Norway,,41.813
Estonia,21.053,Poland,,21.591
Finland,36.646,Portugal,,33.334
France,47.550,Romania,,13.489
Germany,53.358,Slovakia,,18.282
Greece,30.835,Slovenia,,37.970
Hungary,27.692,Spain,,38.873
Iceland,33.801,Sweden,,47.143
Ireland,49.654,Switzerland,,59.902
Israel,59.580,Turkey,,26.250
"",Net Yearly salary,,,Net Yearly salary
Country,average in terms of,,Country,average in terms of
"",PPS,,,PPS
Austria,30.603,Italy,,22.372
Belgium,26.336,Latvia,,18.828
Bulgaria,9.801,Lithuania,,13.507
Croatia,20.254,Luxembourg,,40.942
Cyprus,39.732,Malta,,28.498
Czech Republic,22.252,Netherlands,,35.573
Denmark,24.917,Norway,,26.088
Estonia,13.777,Poland,,14.104
Finland,22.971,Portugal,,21.835
France,26.983,Romania,,12.500
Germany,28.687,Slovakia,,12.173
Greece,24.326,Slovenia,,18.211
Hungary,16.723,Spain,,27.060
Iceland,22.354,Sweden,,22.801
Ireland,28.193,Switzerland,,46.432
Israel,37.389,Turkey,,23.530
```

Figure 15: Output in .xml format

The results, output of extracted tabular data for tested Images can be found out at
https://github.com/strangest-quark/TableExtraction/tree/master/PDFModule

A. We have tested our algorithm on different kinds of Images, PDF's. The entire code, steps for execution of our algorithm is explained step by step in our GitHubRepository.
https://github.com/strangest-quark/TableExtraction

B. Complete Implementation, demonstration of our Algorithm for different types of Input-
https://www.youtube.com/watch?v=H1k2aqCQ1u4&index=15&list=PL3uLubnzL2Tml5Nn3IpDjAc097eTpQevy

6. CONCLUSION AND FUTURE SCOPE

Table detection is through image processing can reduce the hectic process of manual entering of tabular data. It can be mostly used in areas like fin-tech organizations, logistics management and various others fields. For the bank purposes as the files may be confidential and cannot be disclosed, the algorithm can be executed inside the bank's server with strong encryption. This can secure the output files and also reduces the manual work.

The future scope for our work lies in the ability to train the model for certain types of tabular images. Types of tabular images vary across different industries. For example, tables in a bank slip will look different from tables in a logistics slip. Thus the prospect of training the model with different sets of images to improve accuracy for the particular use case adds value to our solution.

REFERENCES

1. Gatos, Basilios, Dimitrios Danatsas, Ioannis Pratikakis, and Stavros J. Perantonis. "Automatic table detection in document images." In International Conference on Pattern Recognition and Image Analysis, pp. 609-618. Springer, Berlin, Heidelberg, 2005. https://doi.org/10.1007/11551188_67
2. Wenzel, Claudia, and Wolfgang Tersteegen. "Precise table recognition by making use of reference tables." In International Workshop on Document Analysis Systems, pp. 283-294. Springer, Berlin, Heidelberg, 1998. https://doi.org/10.1007/3-540-48172-9_23
3. Detect & Correct Skew In Images Using Python | 12 Nov 2016 https://www.google.com/search?q=skew+correction+python&rlz=1C1CHZL_enIN823IN823&oq=skew+correction&aqs=chrome.1.69i57j0l5.4473j0j7&sourceid=chrome&ie=UTF-8
4. pytesseract 0.2.6 | https://pypi.org/project/pytesseract/
5. Hough Line Transform https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_houghlines/py_houghlines.html
6. Erosion and Dilation of images using OpenCV in python |https://www.geeksforgeeks.org/erosion-dilation-images-using-opencv-python/
7. tabula-py 1.3.1| https://pypi.org/project/tabula-py/
8. What is Pipelining? https://www.studytonight.com/computer-architecture/pipelining
9. Morphological Transformations https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html
10. Elmadany, Hassan A., Marco Alfonse, and Mostafa Aref." A Semantic Framework for Summarizing XML Documents", International Journal of Advanced Trends in Computer Science and Engineering." International Journal 6, no. 4 (2017).
11. Chatvichienchai, Somchai, and Yu Kawasaki. "SpreadDB: Spreadsheet-Based User Interface for Querying and Updating Data of External Databases", International Journal of Advanced Trends in Computer Science and Engineering.International Journal 7, no. 2 (2018). https://doi.org/10.30534/ijatcse/2018/01722018
12. Bansal, Sumiti, and Er Rishamjot Kaur. "A review on content based image retrieval using SVM." International Journal of Advanced Research in Computer Science and Software Engineering 4, no. 7 (2014).