This paper is published in Proceedings of Pacific-Rim Symposium on Image and Video Technology (PSIVT) 2015

Table Detection from Slide Images

Xiaoyin Che, Haojin Yang, and Christoph Meinel

Hasso Plattner Institute, University of Potsdam
Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany
{xiaoyin.che,haojin.yang,christoph.meinel}@hpi.de

Abstract. In this paper we propose a solution to detect tables from slide images. Presentation slides are one type of document with growing importance. But the layout difference between slides and traditional documents makes many existing table detection methods less effective on slides. The proposed solution works with both high-resolution slide images from digital files and low-resolution slide screenshots from videos. By taking OCR (*Optical Character Recognition*) as initial step, a heuristic analysis on page layout focuses not only on the table structure but also the textual content. The evaluation result shows that the proposed solution achieves an approximate accuracy of 80%. It is way better than the open-source academic solution Tesseract and also outperforms the commercial software ABBYY FineReader, which is supposed to be one of the best table detection tools.

Keywords: Table Detection, Slide Image, Table Structure

1 Introduction

Table detection is a popular research topic for years. The demand of table detection for numerous of documents, which are stored in libraries, digital archives or on the web, drives the research efforts ahead. But most of these efforts are designed for traditional portrait-oriented, text-dense and book-like documents, which omits one type of frequently used digital document with growing importance in this digitalized new world, the slides.

Presentation slides are widely used in many occasions. The content of slides is vivid, compact and directly focusing on the key points. Therefore "in many organizations, slides also serve a purpose as documentation of information after the presentation has occurred" [1], and special digital library for slides has also been designed [2]. Besides, there are also public slide hosting websites online, such as SlideShare.net, which consists of more than 15 million uploads and is among the top 120 most-visited websites in the world[1]. Furthermore, in the context of education, this never-out-of-date topic, slides play an even more important role. Although whether computer-generated slides are effective in improving the learning outcomes is still under discussion, it is already the basic fact that slides have occupied the front of the classroom nowadays [3, 4]. With the development of

[1] http://www.slideshare.net/about

distance learning, especially the wave of MOOC (*Massive Open Online Course*), huge amount of slides are created and uploaded to the internet as supplementary materials to or being included in the lecture videos everyday. Indexing the tables detected within slides could be helpful in both document retrieval and distance learning contexts.

Unfortunately, the slide layout is very diverse and quite different with traditional documents, which makes lots of existing table detection methods less effective. Thus, we propose a solution to detect table from slide images. The input could be either high-resolution slide images transformed directly from digital files, e.g. PPT or PDF, or the screenshots derived from lecture video with comparatively low-resolution. By taking OCR as initial step, the proposed solution will detect rows and columns, locate the potential table areas and then confirm them by exploring both table structure and textual content.

The rest of the paper is organized as follow: section 2 will introduce the related works and why slide is special, section 3∼5 will illustrate three main technical procedures of proposed solution respectively and then come the evaluation and conclusion.

2 Related Works

The input of an academic table detection approach could be either born-digital PDF files or scanned document images. With the former a solution can extract metadata from the digital files and then do the layout analysis by them [5–7]. With the latter there are different technical solutions, among which ruling line detection [8–10] and whitespace analysis [11, 12] are most popular.

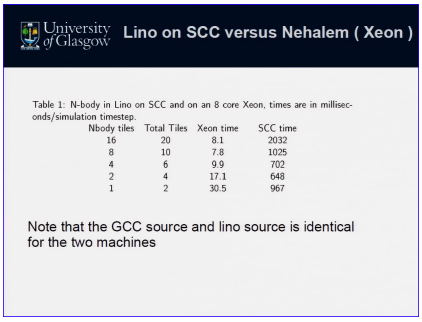
In 2013 a table detection competition was held [13], which enabled all the approaches with either of the above inputs to participate. In addition with 7 academic approaches, the organizer also tested 4 commercial softwares, and the best performer was commercial software ABBYY FineReader 11 with the general accuracy over 98%. Due to the result analysis, the organizer also reported two factors which caused difficulty for most of the approaches: lack of ruling lines and small tables with fewer than five rows. Unfortunately, they are quite common for slide layout, just as Fig. 1-a and Fig. 1-b[2].

There are more specialties in slide layout which may cause problems for table detection effort, such as dark background with light text, diagrams with lines and annotations, sparse but well-aligned 2-columns layout, etc. By applying the updated version of the best performer, ABBYY FineReader 12, on these example slide images, the table in Fig. 1-b is missed and false positive detections are found in Fig. 1-c and Fig. 1-d. Therefore, we believe a table detection method suitable for slides is highly desirable.

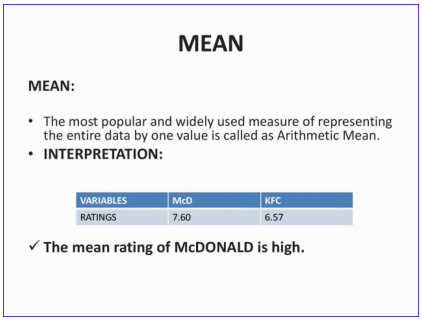
In recent years, some research works also aim for untraditional document layout. Li et al. [14] proposed an approach for particular business forms by recognizing pre-defined header keywords. Ghanmi et al. [15] developed a solution

[2] The copyright belong to original slide authors or institutions: (a)Mr. William Cockshott, (b)Mr. Avi Pipada, (c)Royal Philips Electronics, (d)Ms. Tamara Bergkamp

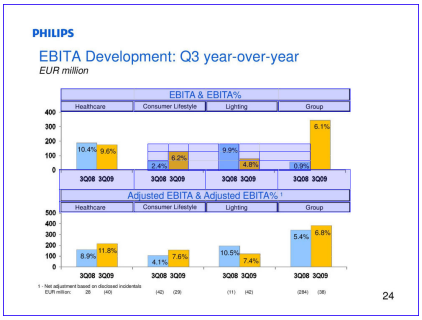
This paper is published in Proceedings of Pacific-Rim Symposium on Image and Video Technology (PSIVT) 2015



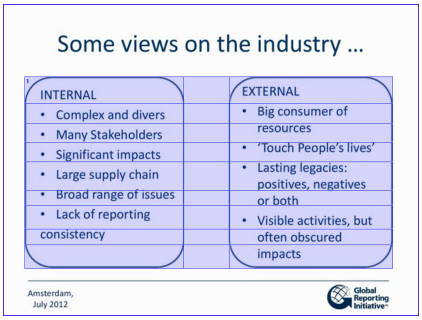
(a) Table without ruling lines



(b) Table with colorful backround and only few cells, missed by FineReader



(c) False positively detected table from a diagram by FineReader



(d) False positively detected table from a 2-columns layout slide by FineReader

Fig. 1. The Challenges of Detecting Table from Slides

for handwritten chemistry document with conditional random fields. And Seo et al. [16] attempted to detect tables from distorted camera-based document image by locating the junctions, which is still relying on ruling lines.

But none of above is suitable for slides. However, the idea of some earlier approaches based on text bounding box clustering [17, 18] could be more inspirational for us. After considering all possibilities, we intend to eliminate all the "shortcuts" and go back to the definition of table: a table is a means of arranging data in rows and columns. To seek row and column structures in slide images will be our initial step.

3 Detection of Rows and Columns

Rows and columns are the indispensable elements of a table. Their existence distinguishes a table from other components in a document, such as a paragraph or a diagram. Therefore, searching and confirming the rows and columns is the premier step of our proposed solution.

After the OCR process, all the textual data within the slide image are stored in text-lines, which include the textual content and the location parameters.

Based on them, a virtual bounding box can be created for each text-line recognized, as shown in Fig. 2-a, and then a slide can be simplified as a bunch of such text-lines and a blank background. As a result, the task becomes to judge whether two text-lines, or we say, two bounding boxes locate in a same row or column.

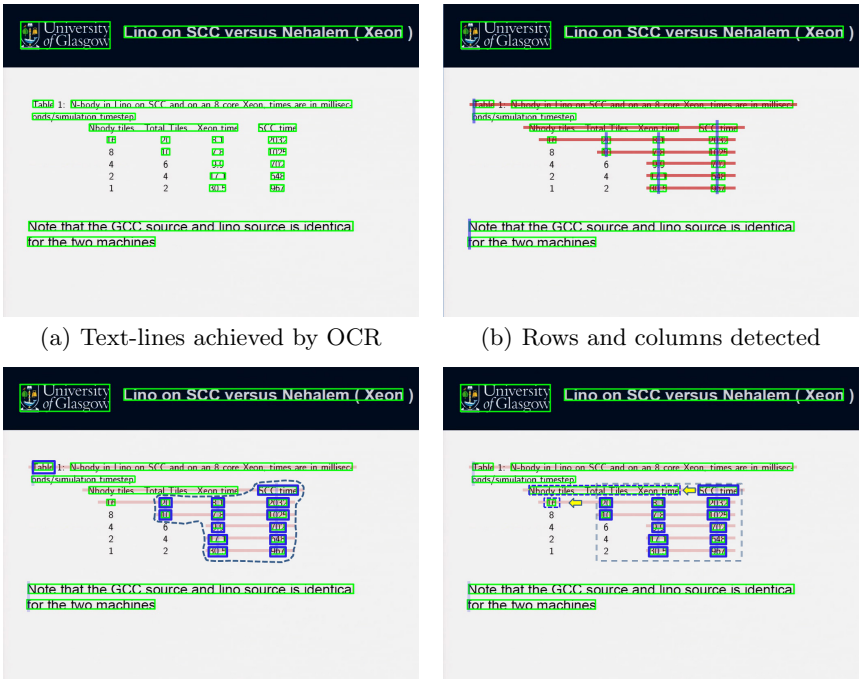
Theoretically it is quite easy to confirm rows. The only requirement is to have two text-lines horizontally locating in a same line. But practically the bounding boxes created for words "Glory" and "name", even with same font, size and actually locating in a same line, may have different heights, and the words "Time" and "map" might even appear interlaced, because of the shapes of letters. In addition with unavoidable and unpredictable OCR errors, a compromised judging mechanism is applied, which requires at least 3/4 of two text-lines vertically overlapp and one cannot be twice the height of the other, or more.

Searching columns is more complicated, with the key issue of alignment. The cells which belong to the same table column must be aligned to the left, to the right or centered. In some special cases, two table cells might coincidently conform to more than one alignment type when they have similar width and same horizontal position. But generally this does not happen to the whole column. Therefore, when we execute the column searching mechanism, one potential column may start with multiple available alignments but end with less as more table cells are involved. The detailed steps are listed below:

1. List all text-lines within current slide as T_1, T_2,..., T_n, and then traverse all possible text-line pairs $T_{ij} = \{T_i, T_j\}, 0 < i < j \leq n$.
2. If T_i and T_j are not vertically aligned, ignore 3∼6 and go directly to 7.
3. If T_i and T_j are vertically aligned, record all their alignment types in A_{ij}. $A_{ij} \subseteq \{Left, Right, Center\}$ and $A_{ij} \neq \emptyset$.
4. Check whether T_i is already included in any existing column candidate C and whether the intersection of A_C (*the alignment types of* C) and A_{ij} is not empty. ($C \subseteq \{T_1, T_2, ..., T_n\}$, $T_i \in C$ and $A_C \cap A_{ij} \neq \emptyset$)
5. If yes, add T_j into C. And set the intersection of A_C and A_{ij} as new A_C. ($A'_C = A_C \cap A_{ij}$)
6. If no, create a new column candidate $C_{new} = \{T_i, T_j\}$. And set $A_{C_{new}} = A_{ij}$.
7. Continue with next pair.

The above mechanism will create quite a lot of false positive table columns, such as a left-aligned text paragraph, a group of annotation in a diagram, or just several unrelated text-lines coincidently seem to be aligned. We will attempt to eliminate these false positive columns in later procedures, but would not risk missing any possibility to find a potential table column here. It is logical to have two columns horizontally overlapped when they belong to different slide components, but they should be vertically separated to each other. If two horizontally overlapped columns are vertically interlaced, or a text-line is shared by two columns, it is most likely to be an error leading by OCR inaccuracy and these two columns will be combined together. Fig. 2-b shows all the 7 rows and 5 columns found in the example slide, including false positive ones.

This paper is published in Proceedings of Pacific-Rim Symposium on Image and Video Technology (PSIVT) 2015



(a) Text-lines achieved by OCR (b) Rows and columns detected

(c) Intersections and the table candidate (d) Table area expansion

Fig. 2. An Example of Table Detection Process

4 Table Area Positioning

4.1 Table Candidate Generation

A table cell is supposed to be the intersection of one row and one column. Rows and columns have already been detected and each of them is a set of several horizontally or vertically aligned text-lines. So the intersected text-lines, each of which belongs to both a row set and a column set, are the most likely table cells. They are the foundation to locate the table area.

Since there might be more than one table in one slide, all the intersected text-lines will be grouped by their rows and columns belonged. Any two text-lines which belong to a same row or column will be grouped together and the effect superimposes. In this way, a text-line is not necessary to share a row or column with all other text-lines in its group but at least share with one of them. Logically the groups should be independent to each other, but when exception happens, which mean one text-line appears in two groups, the two groups will be combined. Any 1-member-group will be directly removed, just like the intersection with text “Table” in left-upper corner of Fig. 2-c.

By now each group represents a potential table candidate. In the following steps they might be modified or eliminated, but never further generated. Any

final table confirmed would evolve from one of these potential table candidates. But first of all, they need to be accepted as formal table candidates by the evaluation based on all available information collected from the intersected text-lines involved, including both textual content and location logic.

4.2 Table Candidate Evaluation

There are four measurements with descending importance implemented in the evaluation process of the table candidate. Content mark and standardized column bonus focus on the textual content of the intersected text-lines, while distance deduction and two-column deduction come from the layout. After accumulating the values of all measurements, only the potential table candidates with positive final value will qualify. In this chapter we only introduce the measurements conceptually, detailed parameter configuration can be found online together with the evaluation datasets.

Content Mark evaluates how likely the content of a text-line looks like the content of a table cell. Although there is no standard or regulation illustrating what can be written in a table and what cannot, people prefer to put numbers, percentages, single-words or short phrases into a table, rather than long sentences. So if the content of an intersected text-line belongs to the first 4 catagories, it earns a positive value. And as the length of the content increases, the content mark decreases to 0 or negative. In the end an average content mark will be calculated, which can be either positive or negative.

Standardized Column Bonus can be only positive or 0. In many cases different table rows are used to identify different subjects, while columns are used to list the values of a certain attribute from these subjects, just like Table 1 and Table 2. When this happens, most cells of a same column contain same type of content, except for the header row. If the content type of a certain column is digit, which includes number, percentage, fraction, etc., or single word, we believe it is a strong evidence to be a table column and earns a big positive value in the evaluation. We set 75% as the threshold to determine whether a column contain same type of content, which allows 1 or 2 cells more than the header line to be the exceptions, especially for the digits, because it is not rare for the OCR to misrecognize 'O' and '0', or 'I' and '1'. When a table is reversely designed, with columns to represent subjects and rows for attributes, standardized column bonus does not work, which is why it has no negative value.

Distance Deduction plays a role when two neighboring columns locating too far away horizontally with each other. It is designed for those slides with chart or diagram, whose description texts are sometimes aligned but remotely located. We take the maximum of 1/8 slide width and the correspondent column's width as the reference. If the gap between two neighboring columns is larger than the reference, a deduction will be applied, and the value of this deduction depends on how much larger the gap is. This measurement can be only negative or 0.

Two-Column Deduction is for the special two-column slide layout, like Fig. 1-d, which is a default layout in most of templates of PowerPoint. When

applied, the items in these two columns are very likely to be aligned both horizontally and vertically, which is quite similar to a two-column table candidate and fairly probable to be detected as one by previous procedure. In order to decrease such false positive detections, if a vertical axis can be found exactly in the middle of the slide to make the two columns symmetric and these two columns contain more than 80% of all text-lines within the whole slide, a two-column deduction will be applied with a comparatively small negative value, because it is only a weak evidence.

4.3 Table Area Expansion

Generally when we talk about a 3×3 table, it should have 9 cells filled with some content. But it is also possible that there are only 6 cells, with 3 in the first row, 2 in the second and 1 in the third. This kind of "triangle" also belongs to table, but obviously some of its cells cannot be involved in any table candidate by previous procedure, because they are not intersections of the rows and columns. And by simply missing some table cells during the OCR process, just like Fig. 2-a, which happens time to time because of the comparatively small size of the text used inside the table cells, some "triangles" or even more weird shapes could be created unexpectedly, such as the "scribbled" table area in Fig. 2-c. No matter in which way it comes, it goes the same, that the area a table candidate can cover is only a part of the real table. In this step, we aim to fix this problem.

In the beginning we draw a virtual rectangle which covers all the text-lines involved in a table candidate and make this rectangle as the initial table area, as shown in Fig 2-d. Then we search potential expansion object alongside the rows and the columns which go across the table area. The content of the targeting text-line and its distance to the current table area will be the decisive factors to judge whether this text-line should be added into the relevant table candidate *(detailed configuration can be found online)*. If so, the table area will be updated. After every expansion to the table area, the whole process will restart until there is no further possibility to include new text-line, which means the area does not change after a full searching process.

5 Table Confirmation

A final confirmation will be made on each table area detected. The measurements of the confirmation process focus not only on cells of the table as what we did in Table Candidate Evaluation, but also consider the table area as a whole. Quite a lot of factors need to be included and they generally form 3 aspects: content, structure and appearance. Since a text block might unlikely "survive" as a false positive table area at this late stage, the main task in the final confirmation is to distinguish those table-like charts or diagrams and eliminate them. One prerequisite is applied before the procedure: if a table area has extreme aspect ratio, such as 10:1 or 1:10, it will also be directly denied because no actual table should be like that.

Content Evaluation involves only one factor, the average content mark, M_c. It is similar to what we did in Table Candidate Evaluation, but the text-lines added in the Table Area Expansion process will also count. Theoretically a bigger value of M_c implies a larger probability of detecting an actual table.

Structure Evaluation involves two factors: scale and integrity. Obviously a table area containing lots cells is more likely to be an actual table. So the total number of table cells detected is the best representative of the scale of the table area, which we address as C_T. And the integrity is also very important. For most tables the expected total cell number C_E should be the product of row number r and column number c within the table area. And the table integrity is defined as the ratio of C_T and C_E, with the range of $(0, 1]$. Both these two factors are positively related with the chance of a table area to be confirmed.

Appearance Evaluation includes two positively related factors, the whole table area A_T and the average text height H_t, and a third factor: text density. A table and its content need to be large enough that people can see it clearly, that is why the A_T and H_t are implemented. And text density is defined as the ratio of the sum of areas covered by all text-lines within the table area and A_T. Addressed as D, the text density of an actual table should be neither too large nor too small. Therefore we set 0.2 as the benchmark by the observation of the ground-truth from the training dataset, and the absolute value of the difference between D and the benchmark 0.2 becomes a negatively related factor.

Now we need to take all these factors together into a general consideration. Theoretically we know the factors are positively or negatively related with the final result, but on practical level, the distribution of each factor's weight is based on the attempt at the training dataset, without mathematical deduction. The final equation of table confirmation can be illustrated as in (1). When the M_{final} is greater than the threshold (*6.5 for slides*), the table will be confirmed.

$$M_{final} = \frac{e^{M_c}}{3} + \frac{\ln C_T \times C_T}{C_E} + \frac{\ln H_t^2 \times (\ln \ln \sqrt[4]{A_T})^3}{e^{\sqrt[3]{|D-0.2|}}} \tag{1}$$

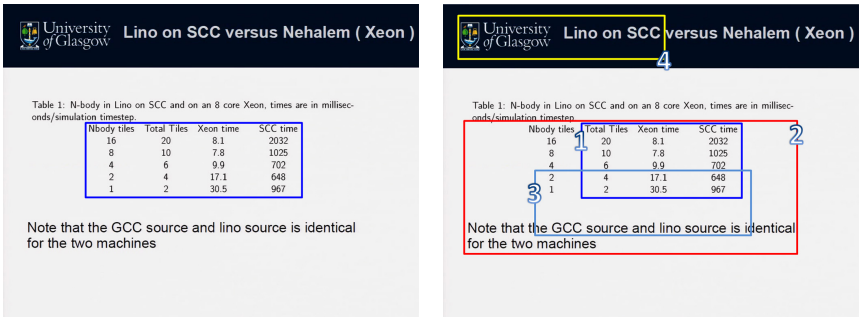
6 Evaluation

6.1 Datasets and Metrics

There are three datasets in our evaluation process: training set, benchmark set and test set. Both training set and test set consist of slide images, which are collected by ourselves, since we failed to find any public dataset of slides for research purpose. The training set gathers 493 slides from 12 complete presentations, which contains 46 tables and is collected from the online learning portal Tele-TASK.de[3]. And we select 384 slides from the excerpts of 26 different presentations on Tele-TASK.de and SlideShare.com as the test set. These slides cover various topics such as economy, education, media, information technology, etc., and contain 189 tables in total.

[3] http://www.tele-task.de/

This paper is published in Proceedings of Pacific-Rim Symposium on Image and Video Technology (PSIVT) 2015



(a) Correct detection (b) Other sorts of detections

Fig. 3. The Catagories of Possible Detection

We intend to compare our proposed solution with the open source Tesseract table detection kit [19] and commercial software ABBYY FineReader 12. In order to avoid misusing these tools, we also implemented a benchmark set of traditional documents, which is shared by the competition organizer of [13] and consisting of 238 pages, including 156 tables. All three datasets can be found online[4].

To evaluate the performances we focus on two facts: how the tables in ground-truth get detected and how accurate an actual detection is. For a table in ground-truth there are 5 possibilities in total: correct detection (*Fig. 3a*), partial detection (*Fig. 3b-1*), over detection (*Fig. 3b-2*), partial-and-over detection (*Fig. 3b-3*) and missed. In actual detections there is an additional false positive category (*Fig. 3b-4*), and in order to quantify the performances, each detected table is given a precision weight as 1, 0.75, 0.5, 0.25 or 0, based on the proportion of its accurately detected area against the ground-truth. Obviously correct detection values 1, false positive (F.P.) values 0, missed table does not have a value while the others depend.

By accumulating the precision values of all actual detections, a recall rate can be calculated against the ground-truth (G.T.) and a precision rate against the number of total detections (T.D.). Please note, that any over detected table will also value 1 when calculating the recall, because in such cases the whole table is actually detected and the extra redundant area will affect in precision. Finally the F_1-Score of the recall and precision will be taken as the general accuracy.

6.2 Experiments on Training and Benchmark Datasets

As we mentioned before, we used the training set to adjust our algorithm. The slide images collected in the training set are directly screenshotted from the desktop stream of lecture videos on Tele-TASK.de. The resolution is 1024×768,

[4] https://drive.google.com/folderview?id=0B13Cc1a7ebTufmhkdzI5VVhSWnotbkhLakh5WVVlVlU2NnlMLVZ2QVpuZDJKdUFyOUtPM1E&usp=sharing

Table 1. Experiments on Training and Benchmark Datasets

Method-Set	G.T.	T.D.	Detection Categories						Recall	Precision	F_1-Score
			Co.	Part.	Over	P&O	Miss	F.P.			
Proposed-T	46	39	21	10	5	0	10	3	69.02%	73.72%	71.29%
Tesseract-B	156	111	69	16	7	2	61	17	55.61%	74.55%	63.70%
FineReader-B	156	153	126	21	0	0	10	6	88.62%	89.87%	89.24%

but the quality of the images is much worse, which causes a lot of OCR errors. Although we aim to cope with OCR errors in the proposed solution, they would still affect in a negative way. The experiment result can be found in Table 1 in "Proposed-T" row, where "Co." and "Part." are the short terms of "Correct" and "Partial" respectively.

The original materials in the benchmark set are saved in PDF format. We transform these files into high quality images for our context. ABBYY FineReader was the best performer, 98% accurate, with the same dataset in [13], when taking born-digital PDF files as input. In our experiment FineReader needs to take the images as input and apply its highly reputable commercial OCR tool [20, 21] on the images while detecting table. As a result, FineReader reaches almost 90% accurate on the benchmark set, which is still very promising. Stats can also be found in Table 1. This result proves the effectiveness of FineReader table detection method on traditional document type. Tesseract works directly with images and its performance on benchmark set can also be referenced.

6.3 Evaluation on Test Dataset

In order to evaluate the proposed solution more comprehensive, we set the slide images in the test set in two different formats: low-resolution screenshot (L) and high-resolution transformed images (H). The resolution of L-images is still 1024×768, for the slide designed in 16:9 ratios there are black edges on the top and the bottom of the slide. But the visual quality of L-images in test set is generally better than those in the training set. We tested all 3 solutions on L-images and the proposed performed the best. H-images are transformed directly from the digital files, either PPT or PDF. They have no unified resolution, but all of them are higher than 1024×768 and visually excellent. All tested solutions perform better on H-images, and the general accuracies of FineReader and proposed solution are almost the same.

From the stats shown in Table 2, we can find out no matter with L-images or H-images, FineReader could achieve more detections than the proposed solution, but also including more false positive detections. The proposed solution tends to make mistakes as over detection, while FineReader is more likely to miss some part of the table. In general, the proposed solution is proven better than FineReader in context of slide images, especially when the input image quality is not so high. On the other hand, Tesseract is no match to either of these two.

By comparing Table 1 and Table 2, it is obvious that both Tesseract and FineReader perform less effective on slide images than traditional type of doc-

Table 2. Evaluation on Test Dataset

Method-Set	G.T.	T.D.	Detection Categories						Recall	Precision	F_1-Score
			Co.	Part.	Over	P&O	Miss	F.P.			
Tesseract-L	189	107	13	34	12	12	118	36	25.93%	41.12%	31.80%
FineReader-L	189	181	109	41	5	2	35	24	73.81%	75.97%	74.87%
Proposed-L	189	169	106	21	26	9	27	7	78.31%	80.18%	**79.23%**
Tessaract-H	189	174	42	42	22	13	74	55	48,81%	47,13%	47,95%
FineReader-H	189	205	142	27	5	0	19	31	84.92%	77.20%	80.87%
Proposed-H	189	194	118	18	32	4	17	22	86.51%	76.03%	**80.93%**

uments, which proves the importance of researching on slide-oriented table detection method. And by analyzing specific instances, we believe our initial aims, such as to avoid recognizing diagram as false positive table, or not to miss the table without ruling line, have been basically fulfilled in the proposed solution. The general accuracy around 80% is not perfect, but enough for some fundamental applications like indexing table-inclusive slides or generating lecture outline.

7 Conclusion

We proposed a table detection method for slide images and achieved quite positive result. Starting with OCR technology, the proposed solution would first detect the rows and columns, locate the table candidates by searching the row-column intersections, expand the areas of these candidates and finally confirm them. The evaluation result shows that the general accuracy of proposed solution is around 80%, which slightly outperforms high reputable commercial software ABBYY FineReader and is way better than open source tool Tesseract. In the future, we would like improve our solution by considering more factors and try our definition-based table detection idea on traditional types of documents.

References

1. Nathans-Kelly, T., Nicometo, C.G.: Slide rules: Design, build, and archive presentations in the engineering and technical fields. Professional Communication, IEEE Transactions on 58(2), 232-235 (2015)
2. Canós, J.H., Marante, M.I., Llavador, M.: Slidl: a slide digital library supporting content reuse in presentations. In: Research and Advanced Technology for Digital Libraries, pp. 453-456. Springer (2010)
3. Hill, A., Arford, T., Lubitow, A., Smollin, L.M.: im ambivalent about it the dilemmas of powerpoint. Teaching Sociology 40(3), 242-256 (2012)
4. Levasseur, D.G., Kanan Sawyer, J.: Pedagogy meets powerpoint: A research review of the effects of computer-generated slides in the classroom. The Review of Communication 6(1-2), 101-123 (2006)
5. Fang, J., Gao, L., Bai, K., Qiu, R., Tao, X., Tang, Z.: A table detection method for multipage pdf documents via visual seperators and tabular structures. In: Document Analysis and Recognition (ICDAR), 2011 International Conference on. pp. 779-783. IEEE (2011)

6. Liu, Y., Bai, K., Mitra, P., Giles, C.L.: Tableseer: automatic table metadata extraction and searching in digital libraries. In: Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries. pp. 91-100. ACM (2007)
7. Yildiz, B., Kaiser, K., Miksch, S.: pdf2table: A method to extract table information from pdf files. In: IICAI. pp. 1773-1785 (2005)
8. Gatos, B., Danatsas, D., Pratikakis, I., Perantonis, S.J.: Automatic table detection in document images. In: Pattern Recognition and Data Mining, pp. 609-618. Springer (2005)
9. Kasar, T., Barlas, P., Adam, S., Chatelain, C., Paquet, T.: Learning to detect tables in scanned document images using line information. In: Document Analysis and Recognition (ICDAR), 2013 12th International Conference on. pp. 1185-1189. IEEE (2013)
10. Tian, Y., Gao, C., Huang, X.: Table frame line detection in low quality document images based on hough transform. In: Systems and Informatics (ICSAI), 2014 2nd International Conference on. pp. 818-822. IEEE (2014)
11. Mandal, S., Chowdhury, S., Das, A.K., Chanda, B.: A simple and effective table detection system from document images. International Journal of Document Analysis and Recognition (IJDAR) 8(2-3), 172-182 (2006)
12. Wang, Y., Phillips, I.T., Haralick, R.: Automatic table ground truth generation and a background-analysis-based table structure extraction method. In: Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on. pp. 528-532. IEEE (2001)
13. Gobel, M., Hassan, T., Oro, E., Orsi, G.: Icdar 2013 table competition. In: Document Analysis and Recognition (ICDAR), 2013 12th International Conference on. pp. 1449-1453. IEEE (2013)
14. Li, J., Wang, K., Hao, S., Wang, Q.: Location and recognition of free tables in form. In: Software Engineering and Knowledge Engineering: Theory and Practice, pp. 685-692. Springer (2012)
15. Ghanmi, N., Belaid, A.: Table detection in handwritten chemistry documents using conditional random fields. In: Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on. pp. 146-151. IEEE (2014)
16. Seo, W., Koo, H.I., Cho, N.I.: Junction-based table detection in camera-captured document images. International Journal on Document Analysis and Recognition (IJDAR) 18(1), 47-57 (2015)
17. Kieninger, T.G.: Table structure recognition based on robust block segmentation. In: Photonics West'98 Electronic Imaging. pp. 22-32. International Society for Optics and Photonics (1998)
18. Shin, J., Guerette, N.: Table recognition and evaluation. In: Class of 2005 Senior Conference on Natural Language Processing (2005)
19. Shafait, F., Smith, R.: Table detection in heterogeneous documents. In: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems. pp. 65-72. ACM (2010)
20. Blanke, T., Bryant, M., Hedges, M.: Ocropodium: open source ocr for small-scale historical archives. Journal of Information Science 38(1), 76-86 (2012)
21. Chattopadhyay, T., Sinha, P., Biswas, P.: Performance of document image ocr systems for recognizing video texts on embedded platform. In: Computational Intelligence and Communication Networks (CICN), 2011 International Conference on. pp. 606-610. IEEE (2011)