Reg.no : 19BCE0903

Name : Vaibhav

Digital Assignment-1

Abstract:

For document analysis table detection is an important part because tables are one of the most effective ways to store information in a systematic manner in a system.There has been a lot of research towards identifying tables not only in documents but even on webpages. In recent times where there are many websites, most of these studies which have been conducted lacks the proper method for table detection,due to variability in size,and shape as image types rather than tags. In this paper, it proposes an efficient and full-bodied method to detect tables in image format, which can be applied to documents and websites. It uses a rule based detection method that utilises key features of many tables, Instead of using recent deep learning methods that require considerable training on large datasets .This proposed method is divided into 2 stage feature extraction ,grid pattern recognition.

Introduction:

Tables are commonly used to deliver relevant information in a structured way to readers because it makes searching, comparing, and understanding information easier. So this is a crucial aspect for document analysis because retrieving broad and well-organized information from documents and websites is an important task.But Table detection, on the other hand, is difficult due to two reasons first considerable intraclass variability and second cross-class similarity amongst tables.A lot of study has been done on table detection over the last few decades. The main topic for its research was the position of the table area. The vast methods used for detecting tables in documents are rule-based,Which focuses on the spaces between text blocks, and objects. In the view of the recent development of technology and trends We use machine learning for generalization purposes because rule-based methods are specific for each document source. Deep learning approaches like CNN have been suggested, and have shown to be effective at detecting websites.

The methods of detecting tables can be divided into two categories.Initially, many studies on the classification of genuine and non-real tables were conducted..We can use mark-up language to detect tables on the websites by analysing HTML. But these tags can also be used for adjusting the layout of the website.Non-real tables are ones that use table tags on a website that aren't part of the actual table, and it's critical to spot them.. The size of the images are large, which leads to increasing the time when applying a stack to the whole image. So it makes in-depth learning approaches unsuitable for crawling data.

Methodology:

In this paper it describes ways for locating tables in photos downloaded from various websites. The technique used in this paper relies on visual signals from website images, it does not use markup language for this purpose.This proposed method is divided into 2 stage feature extraction,grid pattern recognition. In the first stage it extracts features which can be used to represent the contents of a table. This can be done by using text and lines in the image to get each contour's size and position.In first stage we also detect some false positives which are detected outside the table. In the second part the grid pattern is identified by creating a tree structure using the feature collection obtained in the first part. After that we calculate the candidate feature which has features in the grid pattern and is calculated using tree structure obtained above.

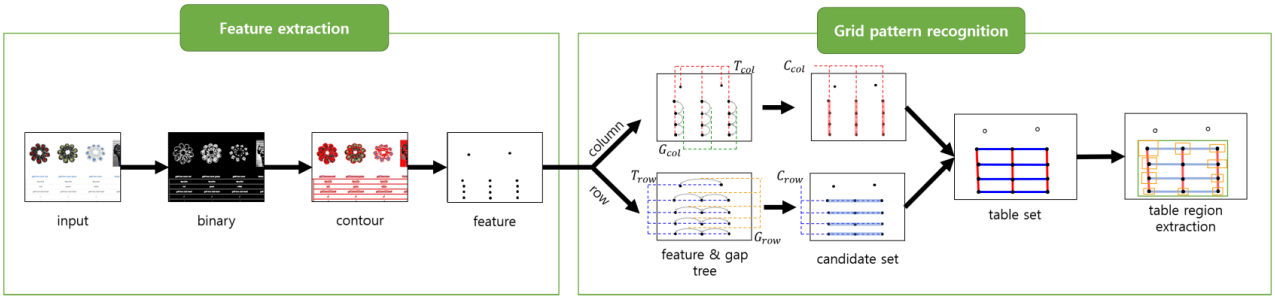


FIGURE 2. **Overview of our method. The method consists of two stages: feature extraction and grid pattern recognition.**

FEATURE EXTRACTION:

First stage determines the position of these features, bounding boxes and the area Which depict the table's contents.We divide these parts into two different parts contour extraction and filtering.Majority of the feature that makes the table are in text format.Due to this reason we firstly extract lines from the area, and after that we remove filter elements which are not text, such as photos or external table.

We use the concept of morphological gradient to extract the image edges. Structural element was used to show the difference between the image inflated and eroding portions. The gradient of image G is calculated from the grayscale image . High frequency components contains both noise and high contrast images.Equation given below

$$G = D(I, K_1) - E(I, K_1),$$

Where

$$D(I, K) = \{z \mid [(\hat{K})_z \cap I] \subseteq I\},$$
$$E(I, K) = \{z \mid (K)_z \subseteq I\},$$

The selected contour still has false positive contours because noisy and high-contrast images have high-frequency components.. In the second half, we use the rule below to try to remove these features from a non-text object.

- Textual features that are legitimate are not duplicated by other features
- Feature area is not empty from inside
- It contains at least least one feature in corresponding level of row and column

GRID PATTERN RECOGNITION :

The table contents or cells are in close proximity to or up and down from each other. So the cell has horizontal and vertical neighbouring cells in rows and columns. Each cell contains its own content in the centre that can be represented to a set of F characteristics. Finding the set of subjects placed in the grid obtained from a set of topics makes it easy to detect the table. We came up with an algorithm that uses the concept of a tree data structure in functions as well as gaps are nodes to address this problem. Initially we constructed 2 trees.We call it sign trees because their nodes are located in the equivalent columns as the child.After that we form Two new trees,which are called feature trees, Which are formed on the basis of the intervals between the parent and its child node.After that from trees, obtained above we select a subset of candidates which can be useful for matching the criteria for deciding table content. Finally, we use the relationship between the two sets of candidates to extract the table areas. The second phase is now divided into three sections.

1. Tree construction

   In this step we constructed ..features.. trees Trow , Tcol by making use of the positions of our features. Our root feature node in each tree takes on the position that corresponds to its own row number, and underneath it we sort by column position for all the rest of the features in that same parent node (recall, our features are ordered from left to right). The subtree of each corresponding tree is a binary tree where its root node becomes either the leftmost or rightmost features located in its row or column parent nodes.

   Features in the tree will be placed at an appropriate depth so they are arranged in descending order according to the column from which they originate. This transformation allows you to easily identify and compare features that you'll find in both columns. Therefore, both Trow and Tcol can be converted into one list of levels for easier navigation and comparison of content.

2. Candidates Set Extraction

   we create sets in a table of contents fashion (you can think of the rows and columns as the top of the table and its contents below) using features and gaps trees. The nodes of either feature or gap trees make up the elements of these sets that follow specific conditions in order to be

considered candidates that may show up in your table. We efficiently determined spacing between content on your website in a manner similar to what occurs in tables. Our findings suggest that it's essential to consider spacing when creating a table of contents; hence we recommend ensuring there's enough space by adhering to standards based on industry benchmarks depending on your type of business, how many people each page can handle at once, etc.

As a result, we develop the following criteria for including a node of the feature tree in a candidate set:

- If consecutive nodes present in gap tree's have similar values, we add the entire gap tree subtree to the candidate set
- If two nodes share similar values in their respective trees, they are combined into a joint node called a Candidate.

The first step is to find characteristics that have a similar spacing in order. The gap tree is used to compare successive gaps of characteristics. The values of two consecutive nodes in a gap tree subtree indicate the spacing of three nodes in the feature trees. For example, we may detect three successive features situated at the same interval when the values of those two consecutive nodes in the gap tree are equal, etc.

3. Extraction of Table Region

Use the candidate sets Crow and Ccol to extract table regions from images.We describe an iterative strategy for expanding the table region from a feature to achieve this goal. We start by picking a node at random C, which is a mix of Crow and Ccol, derived from C.. If a node's child or parent also exists in C, it is added to the table set Bi, where I is the table's index. Because a feature was connected to both feature trees, it became a node during the first phase;The same feature might be put in the union set C as a distinct node., for example, Ta,brow and Tx,y col. Despite the fact that the two nodes represent the same characteristic, they are connected via separate nodes.

We can ensure that a table set is a table set and add index I if it is enlarged into a grid shape. All of Bi's used elements are removed from C. Because an image can include two or more tables, the same process is repeated from another node. In Algorithm 3, A method for extracting a table region is described.When a table has a margin between its contents and its boundary, such as a table surrounded by border lines, we expand the extracted bounding box.Lines cross the bounding box if any pixels in the bounding box have a non-zero value in the binary picture. In this scenario, we move BLT or BRB in out direction till total pixels at the bounding box equals zero

$$B^{LT} = \left(\min_{f \in B} \left(f_x - \frac{f_w}{2} \right), \min_{f \in B} \left(f_y - \frac{f_h}{2} \right) \right),$$

$$B^{RB} = \left(\max_{f \in B} \left(f_x + \frac{f_w}{2} \right), \max_{f \in B} \left(f_y + \frac{f_h}{2} \right) \right).$$

Algorithm 1 for First condition of Grid Pattern :-

Result : C_{row}

Data : Gap tree G_{row}, Feature tree T_{row}

$n \leftarrow$ Degree of Root Node (G_{row})

For $i \leftarrow 1$ to n do

  $j \leftarrow 1$

  While $G_{row}^{i,j+1} \in G_{row}$ do

    $\theta \leftarrow \dfrac{(G_{row}^{i,j} + G_{row}^{i,j+1})}{2\theta}$

    If $|G_{row}^{i,j} - G_{row}^{i,j+1}| < \theta$ then

      $C_{row} \leftarrow C_{row} \cup \{T_{row}^{i,j}, T_{row}^{i,j+1}, T_{row}^{i,j+2}$

    $j \leftarrow j+1$

Algorith 2 :→

The Second Condition of Grid Pattern for C_{row}

Result : C_{row}

Data : Gap tree G_{row}, Feature tree T_{row}, Gap Similarity α, Position Similarity γ

Candidate Set $C_{row} \leftarrow$ DegreeOfRootNode(G_{row})

For $i \leftarrow 1$ to n do

  $G \leftarrow \{ G \mid G^{a,b}_{row} \in G_{row}, a = i \}$

  $S \leftarrow \{ (g^{a,b}, g^{x,y}) \mid\mid (g^{\wedge} - g^{-} \mid < 0, g^{\wedge} \in G^{\wedge} \}$

  While $S \neq \emptyset$ do

    $(g^{a,b}, g^{x,y}) \leftarrow$ GetAnElement(S)

    If $\mid (T^{a,b}_{row})_x - (T^{x,y}_{row})_x \mid < \gamma$ then

      $C_{row} = C_{row} \cup \{ T^{a,b}_{row}, T^{a,b+1}_{row} \}$

    $S \leftarrow S - \{ (g^{a,b}, g^{x,y}) \}$

Algorithm 3 :→

Result : Set of Table feature groups, Tables
Data : Candidate Sets, C_{row}, C_{col}

```
C ← C_row ∪ C_col
i ← 1
While C ≠ 0 do
    T ← ∅
    c ← GetAnElement(C)
    JobStack.Push(c)
    While (JobStack ≠ ∅) do
        P ← JobStack.Pop()
        R ← R − T
        T ← T ∪ R
        While R ≠ 0 do
            q ← GetAnElement(R)
            R ← R − q
    C ← C − T
    dx ← MaxXDistanceOfElement(t)
    dy ← MaxYDistanceOfElement(t)
    If dx > 0 ∧ dy > 0 then
        B_i ← [T]
        i ← i + 1
```

Critical Analysis and Results:

Dataset:

We used two datasets to compare the performance of the document and website images:the TOW image collection and the ICDAR Table Competition dataset.The dataset for the ICDAR (2013) Table Contest consists of 128 PDF documents that describe the placement and organisation of 156 tables. Rather than focusing on one specific subclass,This dataset assumes that the documents were gathered as far as possible.There are 75 tables with papers from the European Union and the United States government.Because we used an image based (table) recognition method, a document page with at least one table is converted to a 200 DPI image.However, for table detection, we continue to use the table bounding box coordinates as the ground truth.

The TOW dataset was created by crawling pictures from a variety of websites, including e-commerce sites and homepages, that contained product specifications.TOW data set have a total of 2,340 pictures and 2,850 tables. The photos of TOW comprise several texts: (English) (Chinese), (Korean) etc as we crawled from many sources.Top-left point and Bottom-right point were used to label the table's bounding box. Bounding box is intuitively matched with the table's boundaries.

Evaluation Metrics:

Because table detection tries to locate a table region, we compare performance using table assessment (.measures.) proposed by ("Gilanietal"), which is on a bounding box region. Bounding boxes are used to show the regions of the ground-truth tables as well as the tables discovered using the method. G denotes a bounding box that has been identified as truth, while "D" denotes "bounding box" discovered by a computer algorithm. Four assessment measures are employed which are used to verify performance: processing time, "(precision)",(" recall, and F1-measure)".

- Precision metric was used assess table detection algorithm's overall performance. It is the percentage of recognised table areas in an image that belong to the real table regions.

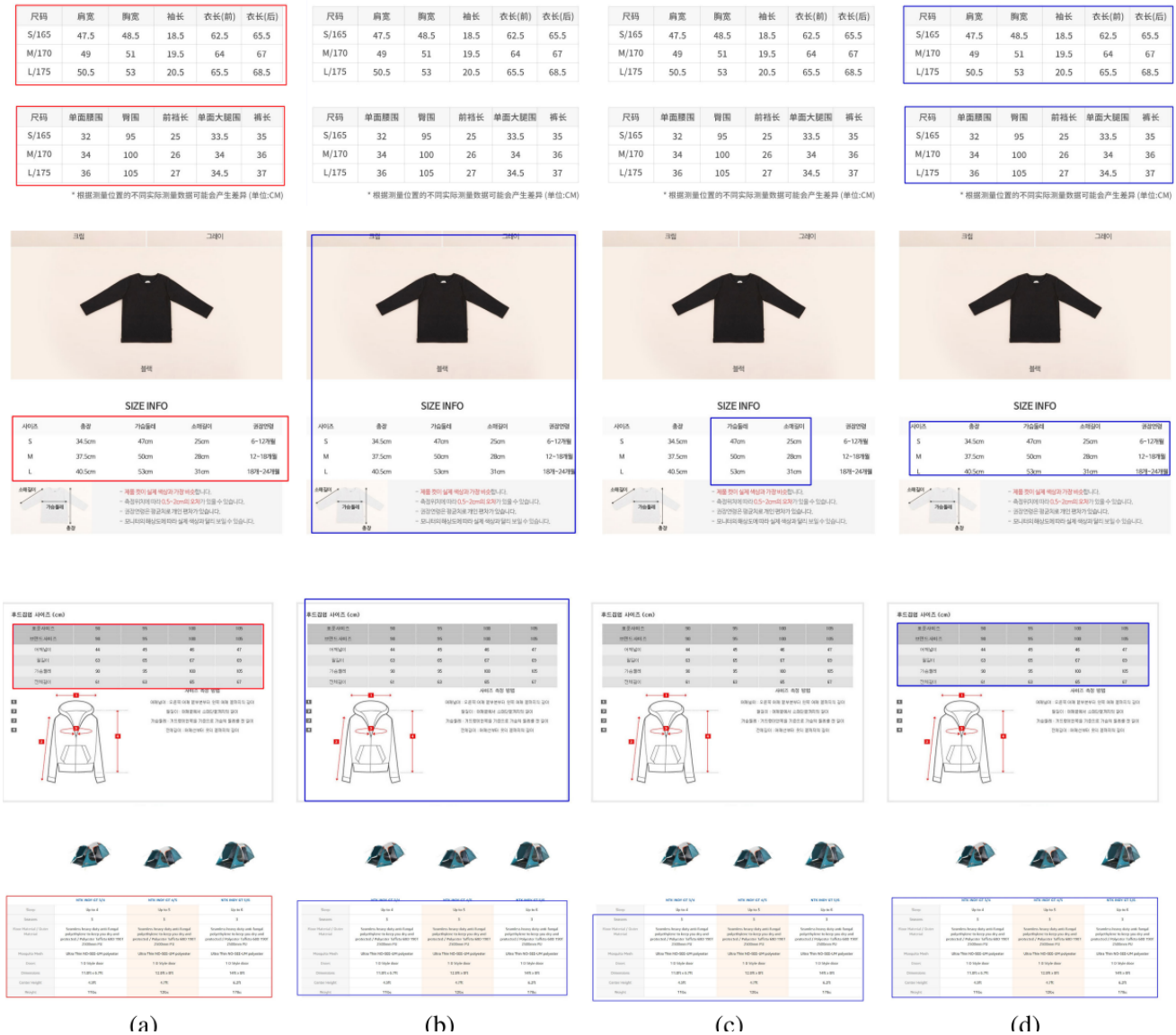
$$\text{Precision} = \frac{|D \cap G|}{|D|}$$

- The recall metric shows the algorithm's detection of ground-truth table sections as a percentage.

$$\text{Recall} = \frac{|D \cap G|}{|G|}$$

- When analysing the methodology's accuracy, the F1 metric strikes a compromise between precision and recall.

$$\text{F1-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$



(a) (b) (c) (d)

Result:

1) RESULT ON ICDAR 2013:

Table 2 compares the performance of TableTrainNet, TableBank, and the proposed technique on the ICDAR 2013 Table competition dataset. The best precision was attained by TableBank, which was 0.9362, which was somewhat greater than the other approaches. TableTrainNet had the best recall and F1-measure, however the detection time was 4-5 times greater than the other approaches. Except for the processing time, which displays the best performance among the several approaches, the suggested method performs somewhat worse than TableBank.

Dataset		TableBank	TableTrainNet	Ours
ICDAR 2013	precision	0.9362	0.9153	0.9148
	recall	0.6376	0.9027	0.5737
	F1-measure	0.7586	0.9089	0.7051
	time (s)	0.7659	3.3495	0.5654

2) RESULT ON TOW:

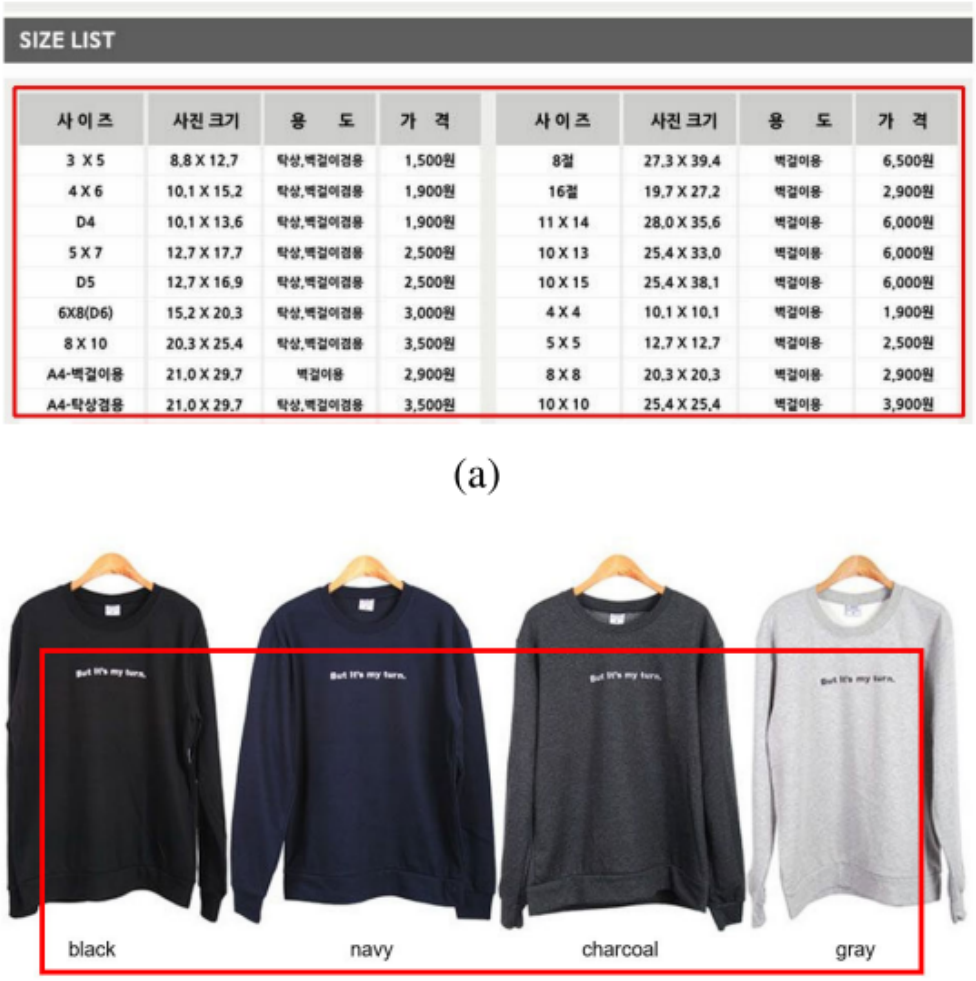
Table 3 summarizes test findings on the dataset of TOW images.Our proposed method clearly outperforms the prior table detection algorithms for website photographs..With both approaches scoring over 0.8, our method has an accuracy of 0.0369 higher than TableTrainNet; however, TableBank's precision drops to 0.5285. Our technique achieves a recall of 0.7206, which is a noticeable difference of 0.2416 from TableBank, which comes in second place in this area. Our technique has the best precision and recall, as well as the best F1-measure of 0.778, which is 0.2755 greater than TableBank.

Table detection requires a rapid processing time since real-world applications demand the handling of enormous numbers of Web site images. TableBank's input size is fixed, which results in a faster calculation time than the other CNN-based approach.

Dataset		TableBank	TableTrainNet	Ours
TOW	precision	0.5285	0.8086	0.8455
	recall	0.4790	0.3003	0.7206
	F1-measure	0.5025	0.4379	0.7780
	time (s)	0.8395	3.4958	0.2816

3)Failure case :

Although we established that robust and reliable recognition of images from documents and webpages is possible, there are limitations. Figure 8 depicts typical failures of our algorithm. When two tables are connected horizontally or vertically without enough space between them, our approach detects them as a single table, as shown in Fig. 8. (a). If the characters are in grid form and appear on both the inside and outside of the image, our approach may interpret them as part of a table.



SIZE LIST

사이즈	사진크기	용 도	가 격	사이즈	사진크기	용 도	가 격
3 X 5	8.8 X 12.7	착상,백겁이겸용	1,500원	8절	27.3 X 39.4	백겁이용	6,500원
4 X 6	10.1 X 15.2	착상,백겁이겸용	1,900원	16절	19.7 X 27.2	백겁이용	2,900원
D4	10.1 X 13.6	착상,백겁이겸용	1,900원	11 X 14	28.0 X 35.6	백겁이용	6,000원
5 X 7	12.7 X 17.7	착상,백겁이겸용	2,500원	10 X 13	25.4 X 33.0	백겁이용	6,000원
D5	12.7 X 16.9	착상,백겁이겸용	2,500원	10 X 15	25.4 X 38.1	백겁이용	6,000원
6X8(D6)	15.2 X 20.3	착상,백겁이겸용	3,000원	4 X 4	10.1 X 10.1	백겁이용	1,900원
8 X 10	20.3 X 25.4	착상,백겁이겸용	3,500원	5 X 5	12.7 X 12.7	백겁이용	2,500원
A4-백겁이용	21.0 X 29.7	백겁이용	2,900원	8 X 8	20.3 X 20.3	백겁이용	2,900원
A4-착상겸용	21.0 X 29.7	착상,백겁이겸용	3,500원	10 X 10	25.4 X 25.4	백겁이용	3,900원

(a)

(b)

FIGURE 8. Failure cases of our method: (a) extracting two side-by-side tables into one, and (b) improperly recognizing characters appearing at regular intervals in the image as patterns.

Conclusion:

Although deep learning-based detection methods have had a huge success in detecting tables They haven't been applied to website photos because they came from document images due to the large amount of Tables come in a variety of shapes and sizes,a scarcity of training data, and extraordinarily huge datasets that take a lengthy time to process. In response to these issues, we suggested Table detection solution based on rules from website photos in this work.We use an image processing method to extract features from the photos, and then we use these features to find grid patterns.

Experiments reveal that our strategy outperforms earlier methods that used deep learning for extraction of features and then used a ".classifier." to website photos.when the method used to recognise tables in documents pictures, it performed somewhat worse than earlier methods but had the fastest processing time. Furthermore, by exploitingthe structures of the table detecting features, the suggested methodology can be used to recover a document to its real form by utilising advantages of rule-based methods. In the future, we want to look at the difficult task of finding tables on complicated photos like our failure examples.