Cell Detection in Tabular Data

A Thesis

submitted to

Indian Institute of Science Education and Research Pune in partial fulfilment of the requirements for the BS-MS Dual Degree Programme

by

Shardul Pramod Manohar

Indian Institute of Science Education and Research Pune

Dr. Homi Bhabha Road,

Pashan, Pune 411008, INDIA.

December 2023

Supervisor: Ajinkya Mundankar

Student Name: Shardul Pramod Manohar

Certificate

This is to certify that this dissertation entitled Segmentation of cells in tabular data towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents work carried out by Shardul Pramod Manohar at AutomationEdge under the supervision of Ajinkya Mundankar, during the academic year 2023.

Ajinkya Mundankar

Committee:

Ajinkya Mundankar

Anindya Goswami

This thesis is dedicated to my Parents

Declaration

I hereby declare that the matter embodied in the report entitled Segmentation of cells in tabular data are the results of the work carried out by me at theAutomationEdge, Pune, under the supervision of Ajinkya Mundankar and the same has not been submitted elsewhere for any other degree

Shardul Manohar

Date: 31/10/2023

Table of Contents

Abstract

The primary goal of our project is to create a **non - deep learning solution** for effectively segmenting cells within tabular data, accommodating tables with or without gridlines.

We have devised an algorithm based on K-Means Clustering to facilitate cell segmentation within tables, irrespective of the presence of gridlines. Our approach involves identifying clusters of characters, often representing words or numbers, and subsequently calculating their centres of mass. We create distinct arrays for the x and y coordinates of these centres. Employing K-Means clustering separately on x coordinates and y coordinates of centres, we determine the optimal number of clusters, denoted as 'k,' from 1 to a predefined maximum value ('max_k') using a novel method for selecting the most suitable 'k', as the existing methods yielded unsatisfactory results. Subsequently, we discern rows and columns separately by employing K-Means clustering with the determined 'k' and identify individual cells through the intersection of these rows and columns.

In addition, we have developed an alternative algorithm tailored for tables containing gridlines. In this scenario, we use canny edge detection and hough transform to detect lines, followed by the identification of intersection points. We use intersection points to detect gridlines. Using these detected gridlines, we reconstruct the table structure.

Acknowledgments

I would like to express my heartfelt gratitude to the individuals who have played an instrumental role in the successful completion of my internship thesis.

First and foremost, I extend my sincere appreciation to my supervisor at AutomationEdge, Ajinkya Mundankar. His expertise, mentorship, and continuous encouragement have been a source of inspiration. His willingness to share his knowledge and insights, as well as his patience in guiding me through the complexities of the internship project, have been pivotal in my personal and professional growth.

I would like to extend my thanks to Pranav Sharma, my senior coworker at AutomationEdge, for his invaluable assistance, support, and camaraderie. His willingness to share his experiences and expertise has been crucial in shaping the direction and quality of this thesis. I am truly grateful for the collaborative spirit and the knowledge I have gained from working alongside him.

Furthermore, I want to acknowledge the invaluable contributions of my college professor, Anindya Goswami, from IISER Pune. His guidance and encouragement have been a source of motivation. His commitment to fostering a deep understanding of the subject matter has been instrumental in shaping my research.

I also extend my gratitude to my family and friends for their unwavering support, understanding, and encouragement during this challenging yet rewarding experience.

Shardul Pramod Manohar

Chapter 1 Introduction

Background and context:

In today's dynamic digital landscape, the processing and analysis of structured data hold a pivotal role in various domains, ranging from business intelligence to scientific research. As Information Technology (I.T.) continues to shape the way we interact with information, the accurate extraction of data from diverse sources becomes crucial for informed decision-making, automation, and enhanced user experiences. In this context, the ability to seamlessly segment tables, detect gridlines etc. emerges as a fundamental challenge with far-reaching implications.

Structured Data and Tabular Representations: Structured data, often presented in tabular formats, constitutes a significant portion of information generated and exchanged in digital environments. Tabular representations are widely utilized in disciplines such as finance, healthcare, engineering, and social sciences for organizing, analyzing, and communicating complex information. However, extracting valuable insights from these tables requires efficient techniques that can handle various data formats, accommodate variations in gridlines, and distinguish between background and content.

Table Segmentation and Gridline Detection: The process of table segmentation involves partitioning a complex table into its constituent cells, facilitating subsequent data analysis and interpretation. Some tables don’t have gridlines so we have to draw gridlines based on on positions of text in it. The tables which do have can be segmented using those gridlines.

Research Gap and Contributions:

We have used K means clustering to segment cells in a table. This algorithm hasn’t been used for this purpose before. We also have

developed a new method for selecting correct K (number of clusters).

Problem statement:

To segment tables both with and without gridlines in constituent cells without using deep learning.

Significance and Impact:

The methodologies and techniques developed in this research hold significant implications for various domains within the field of structured data analysis and Information Technology. By addressing the challenges of table segmentation with and without gridlines while avoiding the complexities of deep learning, this research contributes to the enhancement of data processing efficiency, decision-making processes, and user interaction. The significance of this work can be understood through its potential impact on several fronts.

1. Data Processing Efficiency:

- The developed methodologies enable automated and accurate table segmentation without the need for resource-intensive deep learning techniques.
- Practitioners can benefit from timely data preprocessing and analysis, leading to quicker insights and informed decision-making.
- Efficient table segmentation serves as a foundational step in downstream tasks such as data visualization, analytics, and reporting.
- Non-experts in deep learning can adopt these techniques to effectively process and analyze structured data, democratizing data-driven insights.

2. Real-world Applications:

- The methods presented in this research find applications in various industries, including finance, healthcare, e-commerce, and research.
- Accurate table segmentation is crucial for tasks such as financial analysis, medical record interpretation, inventory management, and data-driven research.

3. Integration into Existing Workflows:

- The non-deep learning nature of the proposed methods facilitates their integration into existing data analysis pipelines and software applications.
- Practitioners can leverage these methods within familiar programming environments, enhancing the capabilities of their tools.

4. Advancement in Image Processing and Analysis:

- The research contributes to the advancement of image processing techniques beyond deep learning paradigms.
- We developed a new method to find the best k for k means clustering.
- By addressing challenges in table segmentation, this work adds to the repertoire of methods available for structured data analysis.

Through the development of practical, efficient, and accurate techniques for table segmentation, this research strives to empower professionals and researchers with tools that streamline data processing, enhance user experiences. By bypassing the computational overhead of deep learning while maintaining high accuracy, these methodologies bridge the gap between research innovation and real-world usability. As we delve into the subsequent chapters, we aim to present the methodologies, experimental results, and implications that underscore the significance and impact of this research.

Literature Review:

Many researchers have worked on this problem. Some of them are.

Smita et al. (2020)[13] proposed an algorithm that “uses a combination of image processing techniques, text recognition and procedural coding to identify distinct tables in same image and map the text to appropriate corresponding cell in dataframe which can be stored as Comma-separated values, Database, Excel and multiple other usable formats.” The algorithm includes binarization of image with Otsu's method and detection of gridlines using contour detection.

Siddiqui et al. (2019)[12] proposed a fully convolutional network (FCN) for table detection. The authors have used a prediction tiling approach based on the consistency assumption of tabular structures. Their method predicts a single column for the rows and a single row for the columns, in order to identify cells. The authors have achieved ‘state-of-the-art' results on the ICDAR-13 image-based table structure recognition dataset. The method utilizes a dual-headed architecture that generates class-specific predictions for rows and columns using a single model, a departure from previous methods that relied on separate models for inference. The results demonstrate that constraining the problem space by imposing valid constraints can lead to significant improvements in performance, with an average F-Measure of 92.39% (91.90% and 92.88% for rows and columns, respectively).

Gatos et al. (2005)[5] proposed a workflow for table detection that “comprises three distinct steps: (i) image pre-processing; (ii) horizontal and vertical line detection and (iii) table detection.” The method detects all intersection points and classifies them according to orientations like four possible corner orientations, four possible T shaped intersections and + shaped intersections and detects and segments tables based on these intersection points.

S.arif et al (2018)[14] depended on the fact that tables contain more numerical data . they used R-CNN deep learning was used for detecting tabular region in the document.

Mandal et al (2006)[8] used the fact that distance between the text in the adjescent column is larger than distance between the words in plain text for table detection.

Nazir et al (2021)[9] used a novel trainable pipeline or neural net called HybridTabNet.

Borra et al (2021)[1] also used faster R-CNN network for table detection

All of these methods required gridlines or deep learning. We were told by the company to develoup a method that doesn't use deep learning and can detect tables without gridlines. Therefore we devloped a K means clustering based algorithm for the purpose.

MacQueen (1967)[7] invented k-means clustering. Which was refined further by Hartigan and wong (1979)[6] however this method is not so far used for table detection

Nobiyuri Otsu[10] in his article proposed a method for selecting threshold for segmenting grey scale images. His method is to select the threshold which minimises the within group variance sum. We use this method to separate text from background.

Chunhui Yuan and Haitao Yang[3] in their article titled "Research on K-Value Selection Method of K-Means Clustering Algorithm" comprehensively investigate and compare four prominent K-value selection algorithms: the Elbow Method, Gap Statistic, Silhouette Coefficient, and Canopy. Their study not only outlines the theoretical foundations of these methods but also provides practical insights by offering pseudo code implementations. Furthermore, Yuan and Yang's experimental validation using the Iris dataset adds empirical rigor to their findings, allowing for a thorough assessment of the advantages and disadvantages of each algorithm.

Calinsky and Harbasz[2] developed the calinsky harbasz index for best k selection.

Peter J. ROUSSEEUW(1987)[11] developed silhouette method of best k selection.

Davies, D. L., & Bouldin, D. W. (1979)[4] introduced the davies Bouldin method for selecting best K.

Tibshirani, R., Walther, G., & Hastie, T. (2001)[15] introduced the gap statistic method for selecting best k.

We tried using these methods for our problem but they didn't properly work so we developed a new method.

Chapter 2 Materials and Methods

1. Table Segmentation Without Gridlines

Theory

A. Otsu's Method for Text Segmentation: Otsu's Method, a widely used image thresholding technique, is designed to automatically determine the optimal threshold value for segmenting objects from the background. Given a grayscale image I and a threshold value t, Otsu's Method aims to minimize the intra-class variance within the object and background regions while maximizing the inter-class variance between them.

The threshold t that optimally separates the two classes can be found by maximizing the inter-class variance V(t):

$$t=\text{argmax}_t\, V(t)=\text{argmax}_t\, \{\omega 0\,(t)\omega 1\,(t)[\mu 0\,(t)-\mu 1\,(t)]2\}$$

Where:

- ω0 (t) and ω1 (t) are the probabilities of the background and object classes, respectively.
- μ0 (t) and μ1 (t) are the mean intensity values of the background and object classes, respectively.

Otsu's Method efficiently identifies the threshold that best separates text from the background, making it suitable for text segmentation in table cells.

B. K-Means Clustering (KMC) for Row and Column Detection: K-Means Clustering is a partitioning algorithm used to group data into K clusters. In the context of table segmentation, KMC can be adapted to detect rows and columns. Given a set of N data points, KMC aims to minimize the sum of squared Euclidean distances between data points and their respective cluster centroids.

The algorithm proceeds as follows:

- Initialize K cluster centroids.
- Assign each data point to the nearest centroid.

- Recalculate centroids based on the assigned data points.
- Repeat steps 2 and 3 until convergence.

By applying KMC to the horizontal and vertical axes of an image containing text, it's possible to detect rows and columns in tables without visible gridlines.

C. Methods for finding Optimal Cluster Number

We tried used the following methods but they did not properly work so we invented a new method.

1. Silhouette Method: The silhouette score quantifies how similar an object is to its own cluster (cohesion) compared to other clusters (separation). A higher silhouette score indicates that the object is well-clustered, while a lower score suggests that it might be in the wrong cluster or that the clustering is not appropriate.

Here's a detailed explanation of the silhouette method:

Silhouette Score Calculation:

For each data point in our dataset, we calculate its silhouette score.

The silhouette score for a single data point is computed using the following formula:

silhouette_score = (b - a) / max(a, b)

where

‘a’ represents the average distance from the data point to other points within the same cluster (intra-cluster distance).

‘b’ represents the smallest average distance from the data point to points in a different cluster (inter-cluster distance).

The silhouette score ranges from -1 to 1:

A score close to +1 indicates that the data point is well-clustered and is far from other clusters.

A score close to 0 suggests that the data point is on or very close to the decision boundary between two neighboring clusters.

A score close to -1 indicates that the data point may be assigned to the wrong cluster.

Silhouette Score for Entire Dataset:

To determine the overall quality of our clustering solution, we calculate the average silhouette score for all data points.

This provides a single value that represents how well the data points are clustered for the given number of clusters.

The number of clusters that yields the highest average silhouette score is considered the optimal number of clusters for our dataset.

A higher silhouette score indicates that the clustering is more appropriate and that the data points are well-separated into distinct clusters.

2. Gap Statistic: The gap statistic is a statistical method used for assessing the quality of a clustering solution by comparing it to a reference or random clustering. It helps in determining the optimal number of clusters for a given dataset. The gap statistic quantifies how much the observed clustering differs from what would be expected by random chance. A larger gap statistic suggests a better clustering solution.

Here's a detailed explanation of the gap statistic:

Clustering Process:

Initially, we have a dataset with data points but no predefined labels or categories.

We apply a clustering algorithm to group the data points into clusters.

Reference Clustering:

To compute the gap statistic, we need a reference or null model.

In the reference model, we generate synthetic data that resembles the original dataset but has no inherent clustering structure.

Within-Cluster Dispersion:

For both the observed clustering (actual clusters) and the reference clustering (random clusters), we calculate a measure of within-cluster dispersion.

Within-cluster dispersion typically quantifies how closely data points within a cluster are grouped together. For example, we can use the sum of squared distances from each point to its cluster centroid.

Gap Statistic Calculation:

The gap statistic is computed as the difference between the observed within-cluster dispersion and the expected (reference) within-cluster dispersion.

Specifically, the gap statistic is calculated as follows:

$$Gap(K) = E[\log(W_{ref})] - \log(W_{obs})$$

K represents the number of clusters we are evaluating.

W_{obs} is the within-cluster dispersion for the observed clustering.

$E[\log(W_{ref})]$ is the expected value of the log of the within-cluster dispersion for the reference clustering. It's typically an average over multiple reference clusterings.

The optimal number of clusters is the one that corresponds to the maximum gap value.

In essence, the gap statistic helps you identify the point at which the clustering solution deviates significantly from random clustering.

Interpretation:

A larger gap statistic suggests that the clustering structure is more pronounced and distinct compared to random chance.

Conversely, a smaller gap indicates that the clustering structure in the data is not significantly different from random.

3. Within-Cluster Sum of Squares (WCSS) Method: WCSS stands for "Within-Cluster Sum of Squares," and it is a metric used to evaluate the quality of a clustering solution, particularly in the context of K-Means clustering. The WCSS measures the compactness or tightness of clusters within a K-Means clustering algorithm. It is used in conjunction with the elbow method to determine the optimal number of clusters for a given dataset.

WCSS Calculation:

For each cluster, calculate the sum of squared distances between each data point in that cluster and the centroid of the cluster. This is the "within-cluster sum of squares" for that cluster.

Mathematically, for cluster 'i', WCSS can be calculated as:

WCSS_i = Σ (distance(data_point, centroid_i))^2 for all data points in cluster i

To get the total WCSS for the entire clustering solution, sum up the WCSS values for all clusters:

Total WCSS = Σ WCSS_i for all clusters

The WCSS method calculates the sum of squared distances between data points and their respective cluster centroids for each value of K. The point where WCSS starts to decrease less rapidly signifies the elbow point.

4. Gap Statistic with Bootstrapping: This extension of the gap statistic incorporates bootstrapping to simulate data variability. Multiple bootstrapped datasets are generated, and the gap statistic

is computed for each. The average gap statistic over the bootstrapped datasets can guide the selection of K.

5. Calinski-Harabasz (CH) Method: The Calinski-Harabasz (CH) method is a statistical method used for evaluating the goodness of clustering in data analysis. It is also known as the Variance Ratio Criterion (VRC) or the index of cluster validity. The CH method is designed to help determine the optimal number of clusters in a dataset when performing clustering algorithms like k-means.

The CH method assesses the quality of a clustering solution by comparing the between-cluster variance to the within-cluster variance. The idea is that a good clustering solution should have minimal within-cluster variance and maximal between-cluster variance. In other words, the CH score is higher when the clusters are well separated and compact.

Here's how the CH score is calculated:

1. Calculate the between-cluster sum of squares (BCSS), which represents the variance between cluster centers. It's the sum of variances between the cluster centroids and the global mean.

2. Calculate the within-cluster sum of squares (WCSS), which represents the variance within each cluster. It's the sum of variances within individual clusters.

3. The CH score is calculated as follows:

$$CH = (BCSS / (k - 1)) / (WCSS / (n - k))$$

where:

- BCSS is the between-cluster sum of squares.
- WCSS is the within-cluster sum of squares.
- k is the number of clusters.
- n is the total number of data points.

The CH score is a measure of how well the data is clustered, and a higher CH score suggests a better clustering solution. Therefore, the goal is to find the number of clusters (k) that maximizes the CH score.

6. Davies-Bouldin (DB) Method: The Davies-Bouldin index (DB index), named after its developers David L. Davies and Donald W. Bouldin, is a clustering evaluation metric used to assess the quality of a clustering solution. It quantifies the average similarity between each cluster and its most similar cluster while considering the compactness of the clusters. Lower DB index values indicate better clustering solutions.

For each cluster 'i', calculate the average distance between all data points in that cluster. This represents the intra-cluster similarity or compactness and is typically referred to as `R_i`.

For each pair of clusters 'i' and 'j', calculate the distance between their centroids (usually Euclidean distance). This represents the inter-cluster dissimilarity or separation and is denoted as `D_ij`.

For each cluster 'i', compute the Davies-Bouldin index `DB_i` as follows:

DB_i = (R_i + R_j) / D_ij

Where 'j' is the cluster that is most similar to cluster 'i' in terms of compactness (lowest `R_i + R_j` and highest `D_ij`).

The Davies-Bouldin index for the entire clustering solution is the average of all `DB_i` values:

DB = (1 / N) * Σ DB_i for all clusters 'i'

A lower DB index indicates a better clustering solution, as it suggests that the clusters are both internally cohesive (low `R_i`) and well-separated from each other (high `D_ij`).

Interpretation:

The Davies-Bouldin index measures the trade-off between cluster cohesion (compactness) and cluster separation (dissimilarity).

Lower values of DB indicate that clusters are well-separated and compact, which is desirable for a good clustering solution.

Values closer to 0 indicate better clustering solutions.

Our algorithm for cell detection

When developing our method, we were instructed by the company to ensure that our model could effectively handle tables without gridlines, all while avoiding the use of deep learning algorithms due to concerns about increased runtime. Therefore, we opted to utilize a K-Means-Clustering based approach to detect tables. The methodology we employed is outlined below:

We will use the example of the following image to explain the steps.

Seed accessions of major candidate species acquired to date in northern Interior B.C.

Plant Family	Species	Biogeoclimatic Subzones Represented	Forest Districts Represented	Subzone x For. Dist. Combinations*	Total Number of Accessions
Asteraceae	*Achillea millefolium*	22	12	42	86
	Anaphalis margaritacea	18	13	39	68
	Antennaria neglecta"	7	7	8	10
	Arnica cordifolia	9	9	15	21
	Aster conspicuus	14	9	15	31
Cyperaceae	*Carex aenea*	13	11	23	38
	Carex macloviana	14	14	25	39
	Carex mertensii	20	15	32	51
Fabaceae	*Lathyrus ochroleucus*	10	10	17	33
	Lupinus arcticus	15	9	23	42
	Lupinus polyphyllus	5	5	8	15
	Vicia americana	11	11	21	35
Juncacoae	*Luzula parviflora*	17	9	18	19
Onaqraceae	*Epilobium latifolium*	11	12	14	17
Poaceae	*Agrostis exarata*	9	6	10	13
	Bromus ciliatus	16	15	22	35
	Calamagrostis canadensis	18	11	29	37
	Calamagrostis rubescens	2	1	2	4
	Danthonia intermedia"	6	6	7	7
	Deschampsia caespitosa"	11	8	11	15
	Elymus glaucus	21	13	46	100
	Elymus innovalus	3	3	3	6
	Elymus trachycaulus	9	7	14	18
	Festuca occidentalis	17	11	31	65
	Festuca saximontana	11	5	12	13
	Poa alpina	8	8	9	11
	Trisetum spicatum	13	10	22	40
Polemoniaceae	*Polemonium pulcherrimum*	4	4	6	6
Rosaceae	*Dryas drummondii*	9	8	10	18
	Geum macophytkim	21	15	23	23
	MEAN:	12.13	9.23	18.57	30.53
(30 species)	TOTAL:	364	277	557	916

Footnotes: * number of combinations divided by 30 expresses the degree of progress in meeting the arbitrary goal of representing 30 different combinations of subzone and district.
** new species not yet established in seed increase plots (1998).

Figure 1 Example Image

Steps:

1) Separate text pixels from the background. For this we used the otsu method for black and white images. For colour images we can use k means clustering with k =2 and use the intensity of R, G, B channels to cluster the pixels in 2 groups and consider the smaller group as text. We put value of 255 for text pixels and 0 for background pixels.
2) Blur the picture so neighboring letters in a word get joined together.
3) We once again apply otsu thresholding.

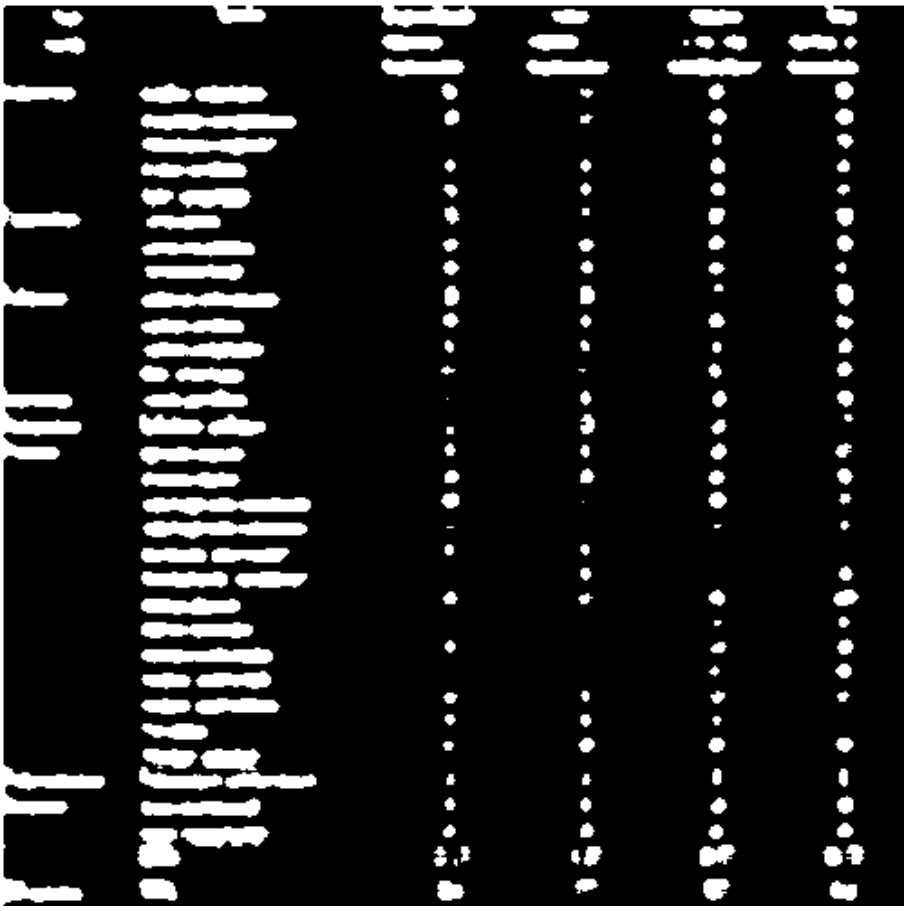


Figure 2 Thresholded image

4) Identify all words in the table and calculate their centers. To do this we identify all directly connected text pixels and calculate the average of each.

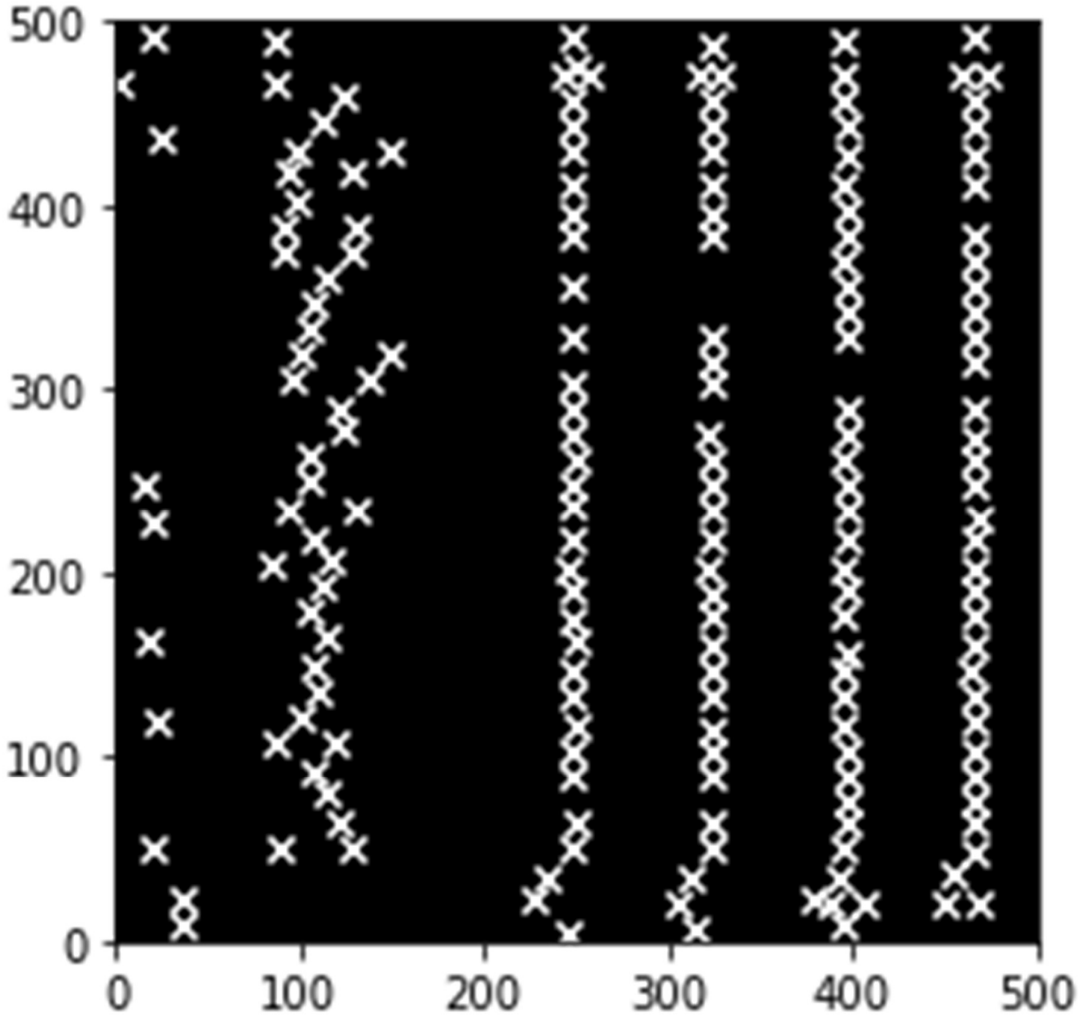


Figure 3 Detected centres

5) Make separate arrays of X coordinates of all centers and Y coordinates of all centers of the words.
6) We will use K means clustering on X coordinates of centers for columns and Y coordinates of centers for rows separately.
7) Calculate the correct number of clusters by the following method. We tried all methods we found on internet to calculate the correct k but they didn't work. So we invented new method to calculate best K as follows. We explain it in more details below.
    1. Make an array of inertias (sum of squared distances of samples to their closest cluster center) for total number of clusters going from number of clusters=1 to number of

clusters = n. (here n is taken as input) call this as array1. (see figure 5)

2. Make another array array2 such that array2[i]=(array1[i-1]+array1[i+1])/(2*array1[i]) for i goes from 2 to n-1 (see figure 7)
3. Make another array array3 such that array3[i]=array2[i-1]+array2[i+1]-2*array2[i] for i goes from 3 to n-2 (see figure 8)
4. Select the number of clusters corresponding to minimum (maximum negative) value in array3. optimum_k=argmin(array3)+1(+1 because index starts at zero instead of1)

8) Fit K means clustering with optimum k calculated above and draw gridlines in the midpoint of consecutive cluster centers.

Plant Family	Species	Biogeoclimatic Subzones Represented	Forest Districts Represented	Subzone x For. Dist. Combinations*	Total Number of Accessions
Asteraceae	*Achillea millefolium*	22	12	42	86
	Anaphalis margaritacea	18	13	39	68
	Antennaria neglecta"	7	7	8	10
	Arnica cordifolia	9	9	15	21
	Aster conspicuus	14	9	15	31
Cyperaceae	*Carex aenea*	13	11	23	38
	Carex macloviana	14	14	25	39
	Carex mertensii	20	15	32	51
Fabaceae	*Lathyrus ochroleucus*	10	10	17	33
	Lupinus arcticus	15	9	23	42
	Lupinus polyphyllus	5	5	8	15
	Vicia americana	11	11	21	35
Juncacoae	*Luzula parviflora*	17	9	18	19
Onagraceae	*Epilobium latifolium*	11	12	14	17
Poaceae	*Agrostis exarata*	9	6	10	13
	Bromus ciliatus	16	15	22	35
	Calamagrostis canadensis	18	11	29	37
	Calamagrostis rubescens	2	1	2	4
	Danthonia intermedia"	6	6	7	7
	Deschampsia caespitosa"	11	8	11	15
	Elymus glaucus	21	13	46	100
	Elymus innovatus	3	3	3	6
	Elymus trachycaulus	9	7	14	18
	Festuca occidentalis	17	11	31	65
	Festuca saximontana	11	5	12	13
	Poa alpina	8	8	9	11
	Trisetum spicatum	13	10	22	40
Polemoniaceae	*Polemonium pulcherrimum*	4	4	6	6
Rosaceae	*Dryas drummondii*	9	8	10	18
	Geum macrophyllum	21	15	23	23
	MEAN:	12.13	9.23	18.57	30.53
(30 species)	TOTAL:	364	277	557	916

Figure 4 Final output

9) Calculate coordinates of cells based on intersection of rows and columns.

Explanation of our best K selection method:

We will us the number of rows in above table as example for our best K selection method. If we plot inertia vs k we get.

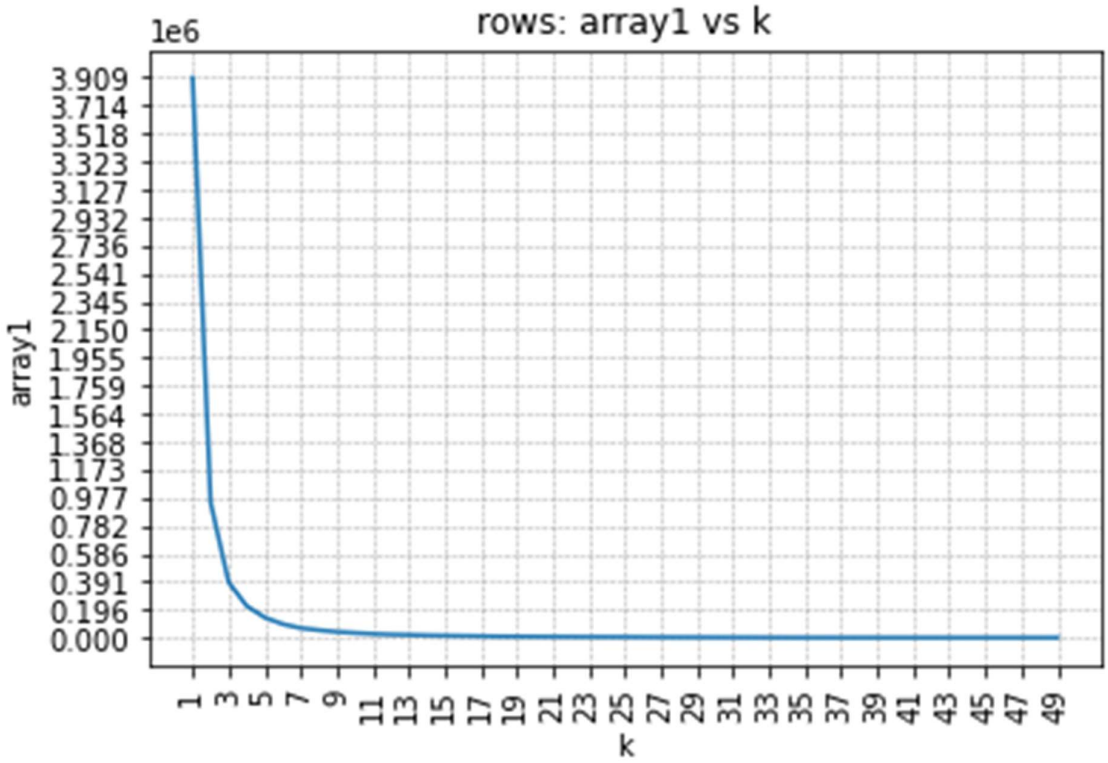


Figure 5 array1 vs k

The correct number for k is 35. The usual method is to select k with the highest double derivative but, the inertia decreases so greatly as k increases that double derivative at 35 will be very low as can be seen in the following graph.

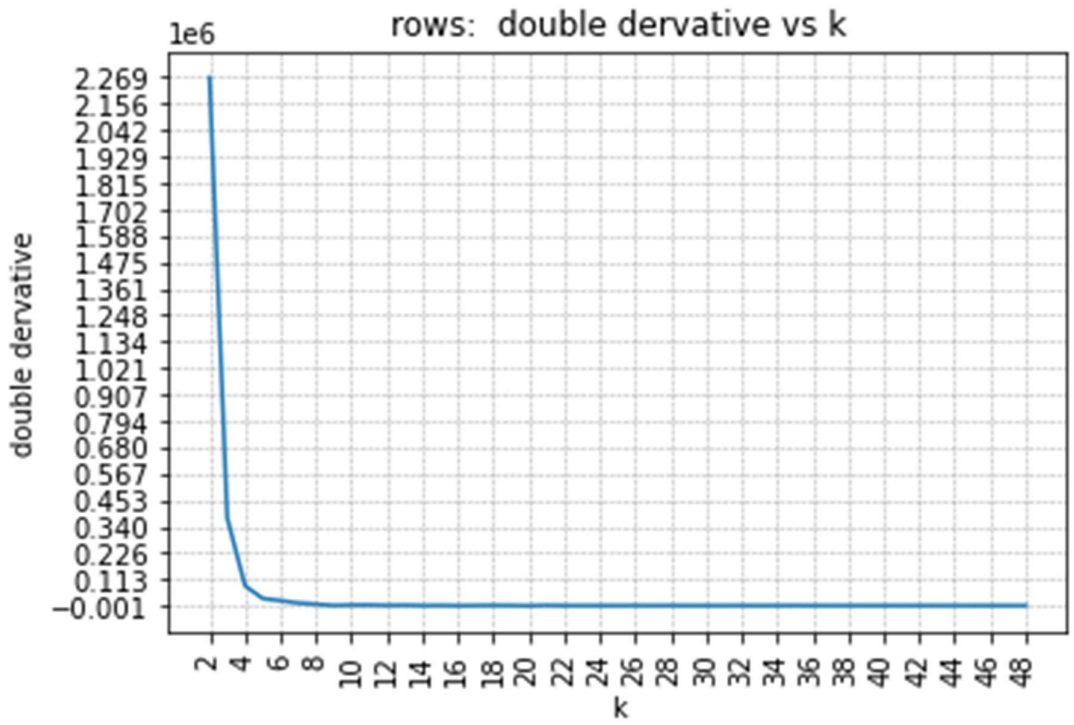


Figure 6 double derivative of inertia vs k

So instead, we use formula array2[i]=(array1[i-1] +array1[i+1])/(2*array1[i]). We get this formula by dividing the double derivative: (array1[i-1] +array1[i+1]) -(2*array1[i]) by 2*array[i] and dropping the constant term. We get the following graph:

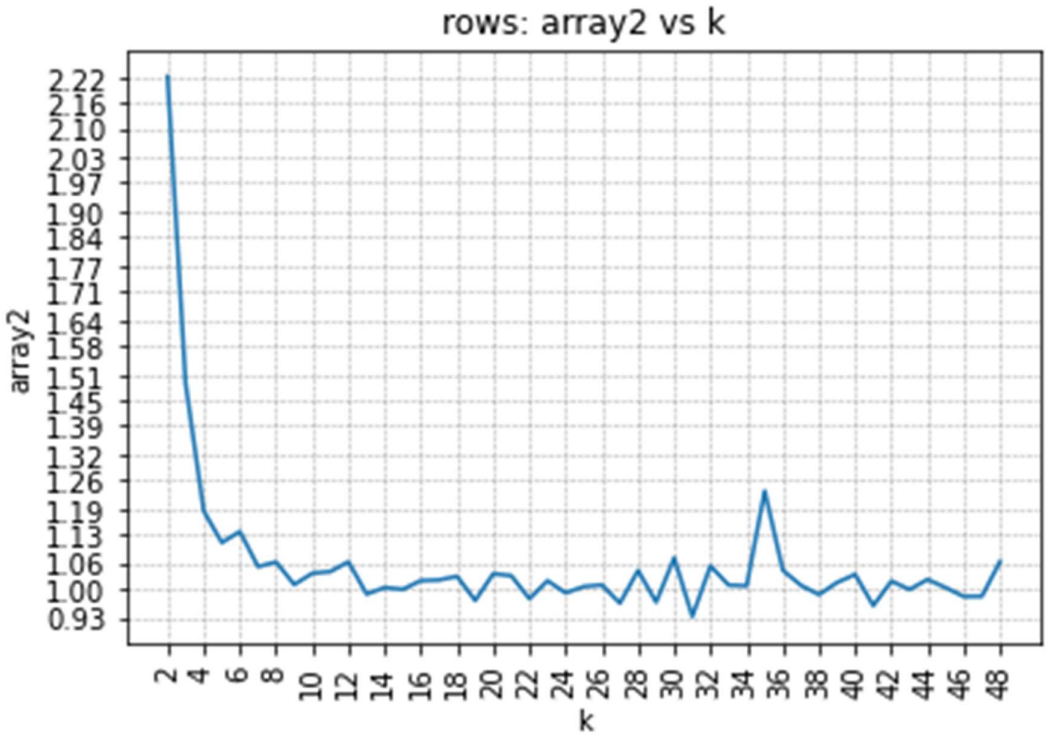
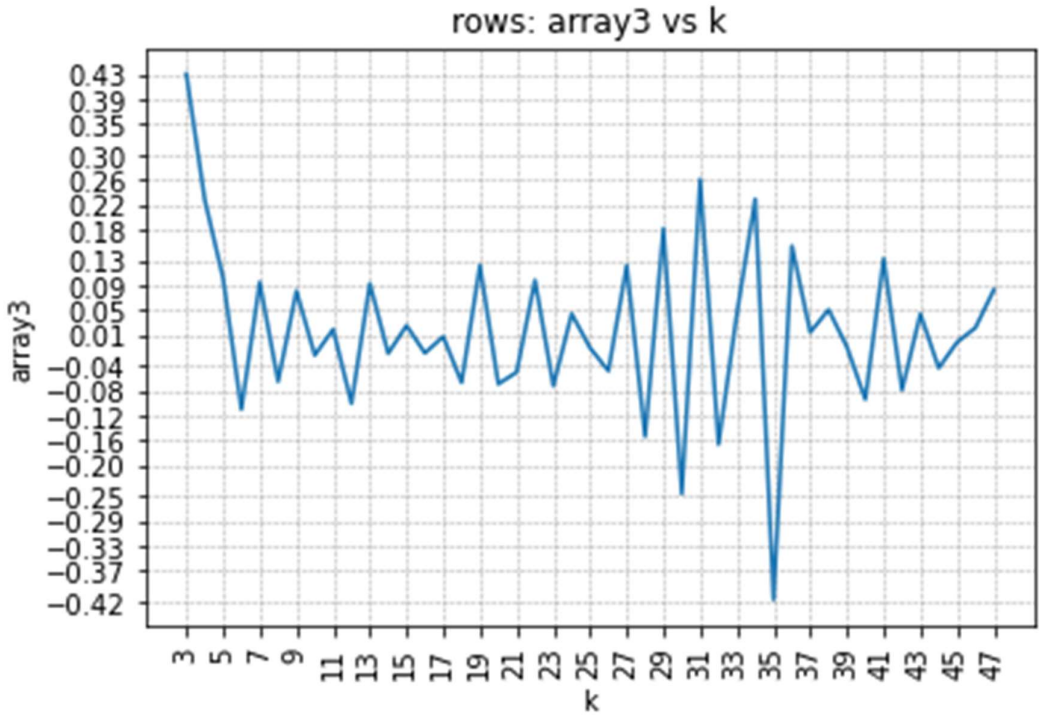


Figure 7 array2 vs k

Here we are getting a sharp peak at 35 as we expected. But we are getting high values at the low numbers of k like 2,3,4 also. So, we take double differential of this and take the most negative value as the correct number of k to find the sharpest peak in array2 as can be seen in the graph below.



.

Figure 8 array3 vs k

2 Table data extraction with gridlines:

For detecting cells in tables with gridlines we use the following method. We first detect the lines using opencv canny edge detection and hough transform functions. Then we calculate the coordinates of all intersection points. We use the rule that if any line has 3 or more intersection points it is a gridline. Then using the identified gridlines

we construct the table. We will use the following image as example

	A	B	C	D	E	F
1	Cookie Sales by Region					
2	SalesRep	Region	# Orders	Total Sales		
3	Bill	West	217	$41,107		
4	Frank	West	268	$72,707		
5	Harry	North	224	$41,676		
6	Janet	North	286	$87,858		
7	Joe	South	226	$45,606		
8	Martha	East	228	$49,017		
9	Mary	West	234	$57,967		
10	Ralph	East	267	$70,702		
11	Sam	East	279	$77,738		
12	Tom	South	261	$69,496		
13						
14						
15						

Figure 9 Example Image

1 Preprocessing

The initial step in our approach involves the preprocessing of the input image. We convert the image into grayscale to simplify further analysis. Grayscale conversion helps in reducing the complexity of the image while preserving essential contrast information.

Figure 10 Greyscale Image

2 line detection

We use canny edge detection to detect the edges in the image. We then use Hough transform to detect lines in the image

3 Intersection Point Detection

The detected horizontal and vertical lines intersect at various points, which serve as the basis for identifying the corners of table cells. Intersection points are critical in defining the geometry of the table structure.

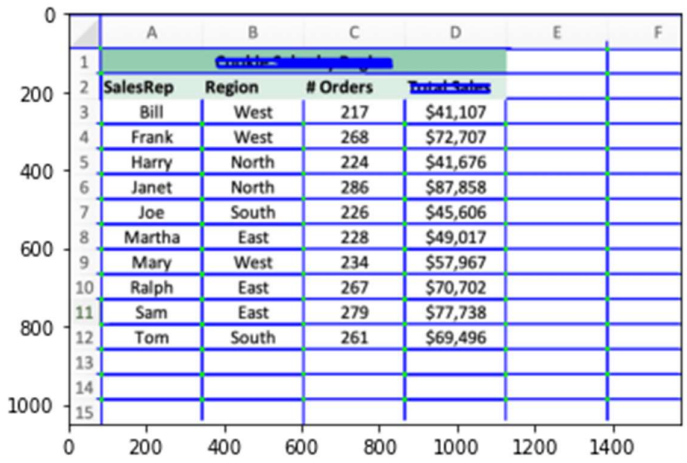


Figure 11 Identified lines and intersection points (yellow dots are intersection points)

4 Line Grouping

4.1 Co-Horizontal and Co-Vertical Linear Points

To identify the gridlines accurately, we group the intersection points that lie on the same line. This grouping is achieved by examining their proximity and orientation. Specifically, we identify sets of points that are approximately co-horizontal or co-vertical.

4.2 Gridline Detection

We consider a line to be a gridline when it contains three or more co-horizontal or co-vertical linear points. The presence of three or more points in alignment suggests the presence of a gridline, which outlines a row or column in the table.

5 Multiple Table Detection

In scenarios where multiple tables are expected in a single screenshot, we employ K-means clustering to separate lines of different tables and segment the cells of each table.

6 Table Reconstruction

Once the multiple tables are detected and the correct number of tables is determined, we proceed to reconstruct each table's structure individually. This involves connecting the identified gridlines to form the boundaries of individual table cells for each table.

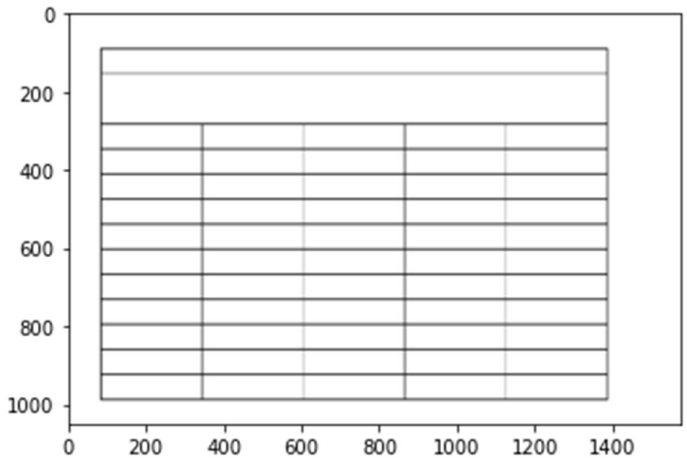


Figure 12 Detected gridlines

Chapter 3 Results

1 Table detection without gridlines: Our program worked well for segmenting tables without gridlines and correctly identified rows and columns most of the time.

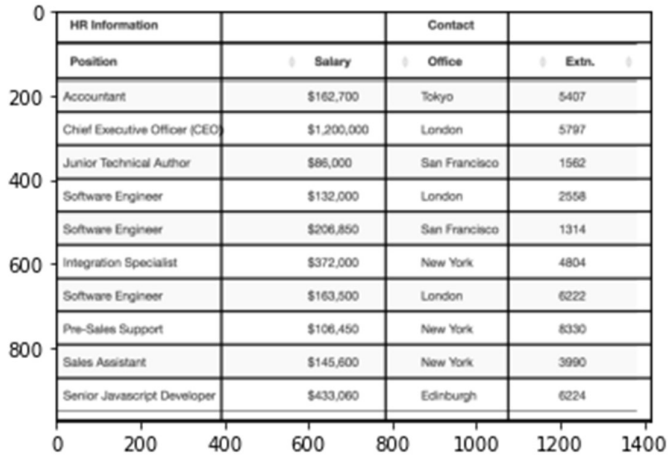
Few examples

Example 1 original image

HR Information		Contact	
Position	Salary	Office	Extn.
Accountant	$162,700	Tokyo	5407
Chief Executive Officer (CEO)	$1,200,000	London	5797
Junior Technical Author	$86,000	San Francisco	1562
Software Engineer	$132,000	London	2558
Software Engineer	$206,850	San Francisco	1314
Integration Specialist	$372,000	New York	4804
Software Engineer	$163,500	London	6222
Pre-Sales Support	$106,450	New York	8330
Sales Assistant	$145,600	New York	3990
Senior Javascript Developer	$433,060	Edinburgh	6224

Figure 13 Example image 1 for table without gridlines

Example 1 segmented image



HR Information		Contact	
Position	Salary	Office	Extn.
Accountant	$162,700	Tokyo	5407
Chief Executive Officer (CEO)	$1,200,000	London	5797
Junior Technical Author	$86,000	San Francisco	1562
Software Engineer	$132,000	London	2558
Software Engineer	$206,850	San Francisco	1314
Integration Specialist	$372,000	New York	4804
Software Engineer	$163,500	London	6222
Pre-Sales Support	$106,450	New York	8330
Sales Assistant	$145,600	New York	3990
Senior Javascript Developer	$433,060	Edinburgh	6224

Figure 14 Segmented image of example image 1 for table without gridline

Example 2 original image

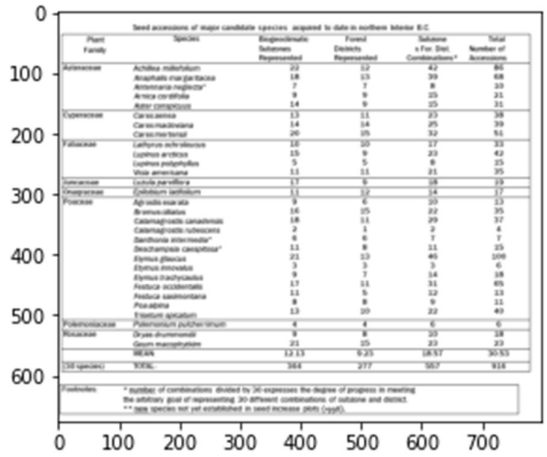


Figure 15 Example image 2 for table without gridlines

Example 2 segmented image

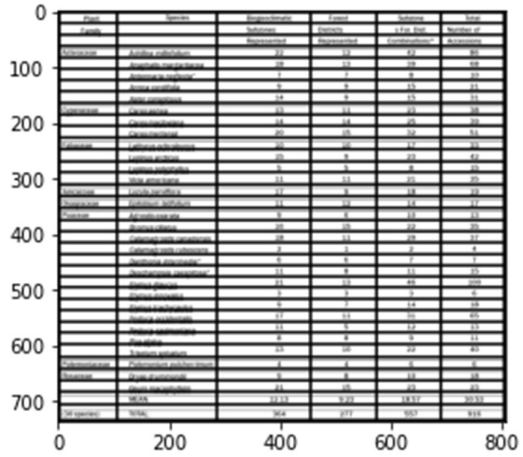
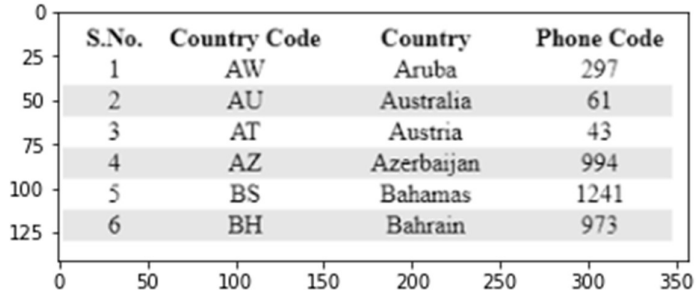


Figure 16 Segmented image of example image 2 for table without gridlines

Example3 original image



S.No.	Country Code	Country	Phone Code
1	AW	Aruba	297
2	AU	Australia	61
3	AT	Austria	43
4	AZ	Azerbaijan	994
5	BS	Bahamas	1241
6	BH	Bahrain	973

Figure 17 Example image 3 for table without gridlines

Example3 segmented image

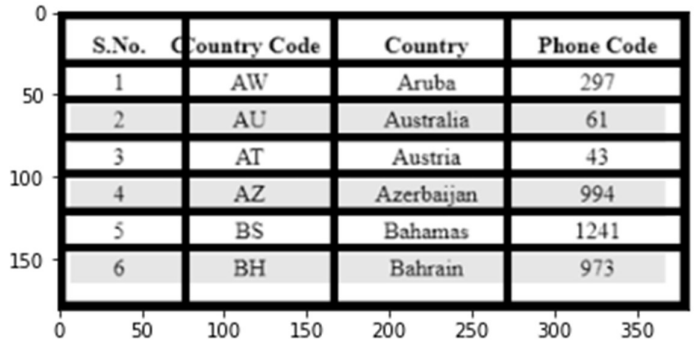


Figure 18 Segmented image of example image 3 for table without gridlines

Limitations(borderline cases):

If the column heading is too wide, and the content of the column is small (ex. 2 digit numeral) then the segmentation based on centre of mass of the column can draw column border that is narrower than the wide heading. Thus cutting through the end parts of column heading.

In a spreadsheet program numerals are usualy right justified and text is usualy left justified, so if a wide column contains both then numerals and text will form separate columns of centre of mass and therefore will be detected as different columns.

Any dark coloured (text coloured) object will be considered part of text and its centre of mass will be calculated and considered while fitting the kmc.

> Remedy : Objects which are obviously too big (like gridlines) to be considered text can be removed. We have used this in our program. However small non text artifacts will still cause inaccuracies.

2 Tables with gridlines: Our program worked well most of the time and also could handle merged cells.

Example 1 original image

	A	B	C	D	E	F
1	Cookie Sales by Region					
2	SalesRep	Region	# Orders	Total Sales		
3	Bill	West	217	$41,107		
4	Frank	West	268	$72,707		
5	Harry	North	224	$41,676		
6	Janet	North	286	$87,858		
7	Joe	South	226	$45,606		
8	Martha	East	228	$49,017		
9	Mary	West	234	$57,967		
10	Ralph	East	267	$70,702		
11	Sam	East	279	$77,738		
12	Tom	South	261	$69,496		
13						
14						
15						

Figure 19 Example image 1 for table with gridlines

Example 1 detected gridlines

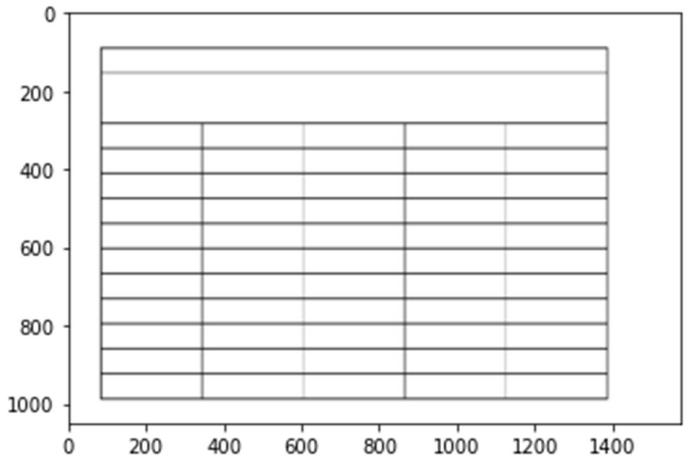


Figure 20 Detected gridlines of example image 1 for table with gridlines

Example 2 original image

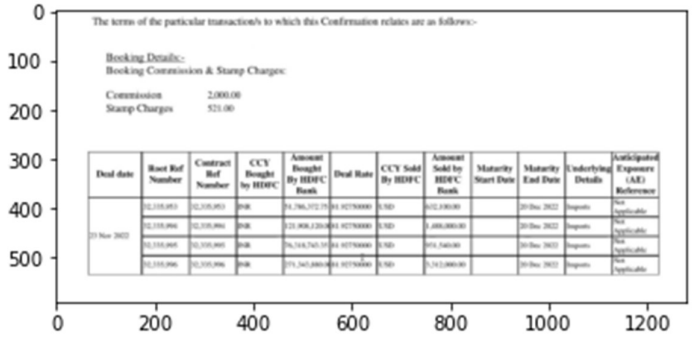


Figure 21 Example Image 2 for table with gridlines

Example 2 detected gridlines

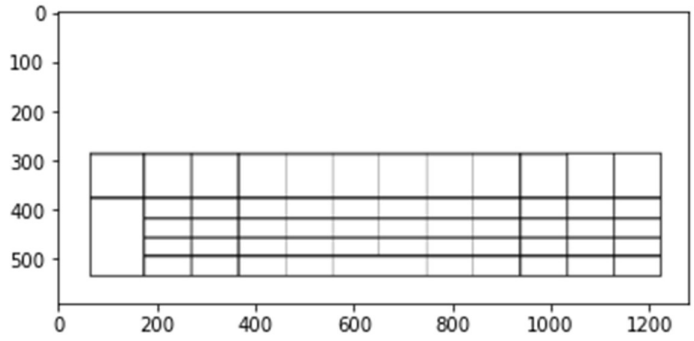


Figure 22 Detected gridlines of example image 2 for table with gridlines

Chapter 4 Discussion

Our program has effectively detected cells within the majority of the sample images provided by the organization to which I am currently affiliated. It is noteworthy that while a substantial body of scholarly literature exists pertaining to the segmentation of tables, the utilization of the K-means clustering methodology for table segmentation remains conspicuously absent in extant research works. Our introduced method for finding best K, which is an original contribution, has demonstrated superior performance in comparison to pre-existing techniques, particularly within the context of the present problem.

It is noteworthy that a predominant number of extant scholarly papers primarily rely on either gridlines or deep learning techniques for table segmentation. It is imperative to emphasize that, based on directives from our company's leadership, the employment of deep learning approaches was explicitly prohibited due to runtime constraints. Consequently, we have innovatively devised a novel methodology capable of accurately segmenting tables, irrespective of the presence or absence of gridlines, without necessitating the use of deep learning techniques.

Additionally, we have developed a separate software program tailored for the precise segmentation of tables that incorporate gridlines. This specialized program also exhibits the ability to adapt to scenarios involving merged cells and multiple tables within the same document.

References

[1] Borra Vineetha, D. N. D., and Ravi Yelesvarupu. "Automatic Table Detection, Structure Recognition and Data Extraction from Document Images."

[2] Caliński, Tadeusz, and Jerzy Harabasz. "A dendrite method for cluster analysis." *Communications in Statistics-theory and Methods* 3, no. 1 (1974): 1-27.

[3] Yuan, Chunhui, and Haitao Yang. "Research on K-value selection method of K-means clustering algorithm." *J* 2, no. 2 (2019): 226-235.

[4] Davies, David L., and Donald W. Bouldin. "A cluster separation measure." *IEEE transactions on pattern analysis and machine intelligence* 2 (1979): 224-227.

[5] Gatos, Basilios, Dimitrios Danatsas, Ioannis Pratikakis, and Stavros J. Perantonis. "Automatic table detection in document images." In *Pattern Recognition and Data Mining: Third International Conference on Advances in Pattern Recognition, ICAPR 2005, Bath, UK, August 22-25, 2005, Proceedings, Part I 3*, pp. 609-618. Springer Berlin Heidelberg, 2005.

[6] Hartigan, John A., and Manchek A. Wong. "Algorithm AS 136: A k-means clustering algorithm." *Journal of the royal statistical society. series c (applied statistics)* 28, no. 1 (1979): 100-108.

[7] MacQueen, James. "Some methods for classification and analysis of multivariate observations." In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14, pp. 281-297. 1967.

[8] Mandal, Sekhar, S. P. Chowdhury, Amit K. Das, and Bhabatosh Chanda. "A simple and effective table detection system from document images." *International Journal of*

Document Analysis and Recognition (IJDAR) 8, no. 2-3 (2006): 172-182.

[9] Nazir, Danish, Khurram Azeem Hashmi, Alain Pagani, Marcus Liwicki, Didier Stricker, and Muhammad Zeshan Afzal. "HybridTabNet: Towards better table detection in scanned document images." *Applied Sciences* 11, no. 18 (2021): 8396.

[10] Otsu, Nobuyuki. "A threshold selection method from gray-level histograms." *IEEE transactions on systems, man, and cybernetics* 9, no. 1 (1979): 62-66.

[11] Rousseeuw, Peter J. "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis." *Journal of computational and applied mathematics* 20 (1987): 53-65.

[12] Siddiqui, Shoaib Ahmed, Pervaiz Iqbal Khan, Andreas Dengel, and Sheraz Ahmed. "Rethinking semantic segmentation for table structure recognition in documents." In *2019 international conference on document analysis and recognition (ICDAR)*, pp. 1397-1402. IEEE, 2019.

[13] Pallavi, Smita, Raj Ratn Pranesh, and Sumit Kumar. "A conglomerate of multiple OCR table detection and extraction." *arXiv preprint arXiv:2010.08591* (2020).

[14] Arif, Saman, and Faisal Shafait. "Table detection in document images using foreground and background features." In *2018 Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1-8. IEEE, 2018.

[15] Tibshirani, Robert, Guenther Walther, and Trevor Hastie. "Estimating the number of clusters in a data set via the gap statistic." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63, no. 2 (2001): 411-423.