Vertical Bar Chart Comprehension

Tai An, Bowen Yang
Department of Computer Science, University of Rochester, New York 14627, USA
(Dated: May, 2023)

Vertical bar charts are ubiquitous in various fields as a tool for representing and visualizing data. Accurate extraction of information from these charts, including x-axis values, y-axis values, and the corresponding values of each bar, is critical for data analysis and decision-making. This paper presents a novel hybrid approach that combines rule-based methods and deep learning techniques to achieve high-accuracy information extraction from vertical bar charts. Our method utilizes deep learning for OCR text extraction, axes detection, and bar detection, while relying on a rule-based system for text classification and data inference. We conducted extensive experiments to evaluate the performance of our proposed approach on a diverse dataset of vertical bar charts from different sources and domains. The results demonstrate that our hybrid approach achieves an outstanding accuracy around 99%. This high level of accuracy has important implications for the automation of data extraction and analysis from vertical bar charts, paving the way for more efficient and accurate decision-making in various domains, including finance, education, and business analytics.

I. INTRODUCTION

Charts are commonly used to present data in all kinds of documents, books and papers. When the underlying data is not available, it's necessary to extract the data from the charts. But when it comes to those with disabilities, extracting information from charts can be difficult or even impossible. Millions of students have a learning, physical, or visual disability that prevents a person from reading conventional print. The majority of educational materials for science, technology, engineering, and math (STEM) fields are inaccessible to these students. Technology that makes the written word accessible exists. However, doing so for educational visuals like graphs remains complex and resource intensive. As a result, only a small fraction of educational materials are available for learners with this learning difference unless machine learning could help bridge that gap. The goal of this competition is to extract data represented by four types of charts commonly found in STEM textbooks. In this project, we focus on dealing with the data extraction for vertical bar charts which are frequently used to represent data sets and reveal trends or patterns in a visually intuitive manner.

Like Figure 1 has shown. A Vertical bar chart typically consists of several key components that work together to convey information. The x-axis represents the categorical variables or discrete data points, while the y-axis corresponds to the numerical values or continuous data points. Each vertical bar on the chart represents a data point or category, with the height of the bar indicating the corresponding value. In addition to the axes and bars, vertical bar charts often include labels for both axes, as well as individual data points or categories. These labels provide context and facilitate the interpretation of the data being visualized.

The existing method, ChartReader[5], uses a deep learning based classifier to determine the chart type of a given chart image and novel heuristic methods for analyzing scientific figures (text detection, pixel grouping, object detection) and address prime challenges like axis detection, legend parsing, and label detection. However, our proposed method for vertical bar chart comprehension leverages the strengths of both deep learning techniques and rule-based systems. The deep learning component of our approach is responsible for OCR text extraction, axes detection, and bar detection. OCR text extraction enables the accurate identification and extraction of textual information from the charts, while axes and bar detection identifies the key structural elements of the charts. On the other hand, the rule-based system is employed for text classification, then combined with location of detected bars, data inference can be performed.

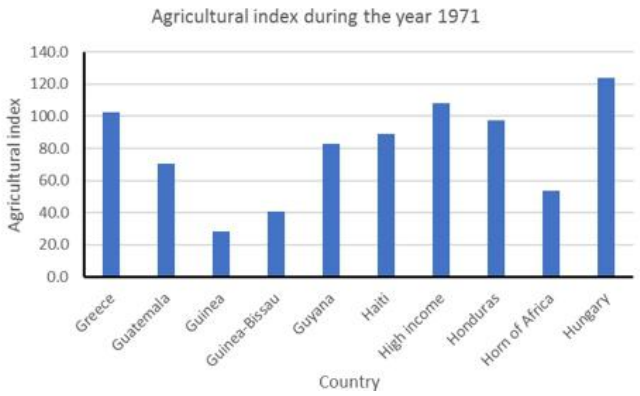

FIG. 1. Example of Vertical Bar Chart .

II. RELATED WORK

Accurately parsing bar-plots in scientific literature presents a significant challenge due to the wide variety of plot types. Prior work has concentrated on constructing heuristic models and rule-based methods that identify essential features, such as bars and x-axis and y-axis semantics, eg.,[8][9]. The works by[1] and[2] first classify

the chart type using a Support Vector Machine, which is then followed by the use of heuristic based data extraction methods. Such rule-based methods have become less prominent due to the lack of generalisability to real-world data and the need to hand-engineer features[6]. In the case of Zhou et al. (2001) [10], ergodic hidden Markov models were used to identify bar-plots by leveraging feature extraction for parameter estimation within the model. Likewise, Davila et al. (2019) [11] made use of PixelLink text detection and SVM training to link feature labels to their respective components. However, neither of these studies focused on extracting data values from the bar-plots.

Recently, efforts in deep learning based models have been shown to give better results in chart data extraction. These methods usually build upon object detection models and adapt the processing according to the chart type. ChartText[3] uses a Convolutional Neural Network to classify the chart type, then runs object detection models to extract text and labels. Then, type-specific image processing is done to extract the data. ChartOCR [4] also uses a CNN to classify the chart type, which is then followed by a keypoint detection model to extract certain pre-defined keypoints based on the chart type. Postprocessing is done based on the detected chart type to extract chart data using the identified keypoints. These methods perform better than rule-based methods, but they still give problematic results on real-world data.

In [7] the authors propose a method for extracting numerical data from scatter plots using a combination of image processing techniques and machine learning. The method involves several steps, such as preprocessing, axis detection, tick mark detection, tick label extraction, point detection, point coordinate extraction, and post-processing. However, the method has some disadvantages, including sensitivity to noise, dependence on OCR accuracy, difficulty in handling complex plots, limitations in handling different plot styles, and the need for parameter tuning. These challenges can affect the overall performance of the algorithm and the accuracy of the extracted data.

Despite the advancements in automated bar chart parsing, existing methods still face challenges in dealing with variations in chart design, noise, and occlusion. Our proposed hybrid approach, which combines deep learning for OCR text extraction, axes detection, and bar detection with a rule-based system for text classification and data inference, aims to address these shortcomings while achieving high accuracy in information extraction from vertical bar charts.

III. DATASET

To evaluate the performance of our hybrid approach for vertical bar chart comprehension, we used a diverse dataset containing a variety of vertical bar charts in competition datatset collected from scientific publications, news articles, papers, and websites. Our dataset consists of 3168 vertical bar chart images, where each image is accompanied by a set of annotations. The images in the dataset represent a wide range of vertical bar chart styles, including single and grouped bars, as well as charts with various color schemes, axis styles, and labeling conventions. This diversity ensures that our method is tested on a broad spectrum of real-world scenarios, enabling a more comprehensive evaluation of its performance and robustness.

Each image in the dataset was annotated with the following information:

- Chart Title: The position of title in the image.
- X-axis label: The position of X-axis label in the image.
- Y-axis label: The position of Y-axis label in the image.
- X-axis values: The tick labels and corresponding positions along the X-axis.
- Y-axis values: The tick labels and corresponding positions along the Y-axis.
- Bar values: The value associated with each bar, along with its X-axis values.

The annotated dataset was then split into training, validation, and test sets, following the principle of 7:2:1 ensuring a balanced representation of different vertical bar chart styles in each set. The training and validation sets were used to fine-tune and validate our hybrid approach, while the test set was reserved for the final evaluation of the method's performance.

In order to perform bar detection, we manually labeled 300 images for fine-tuning YOLOv5, and also split into training set (209 images), validation set (59 images), and test set (32 images), also following the ratio 7:2:1.

IV. METHOD

A. OCR Texts Extraction

We developed a double-pass algorithm that utilizes the AWS-Rekognition DetectText API for detecting text in a figure, with the goal of enhancing text detection results. DetectText not only provides rectangular bounding boxes for the detected text but also outputs a confidence score, indicating the likelihood of a correct prediction. Algorithm 1 illustrates the pseudo-code for our double-pass algorithm.

Our algorithm is based on the premise that by whitening out high-confidence detected text, we can minimize interference and improve the quality and confidence scores of other text elements with initially lower scores. To establish an appropriate confidence score threshold,

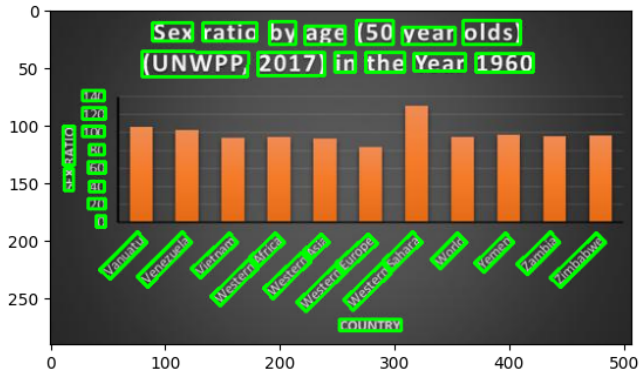

FIG. 2. Illustration of Text Extraction.

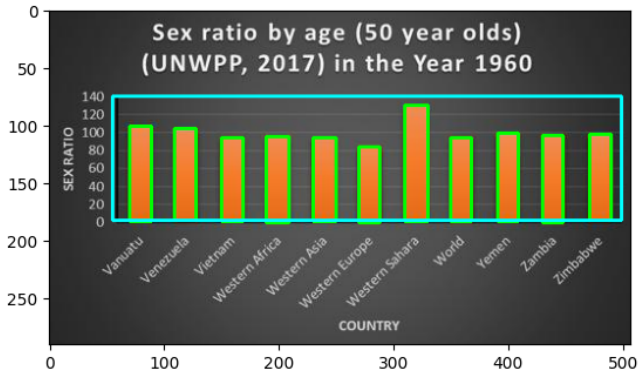

FIG. 3. Bounding boxes of plot and bars.

we examined the text detection results of a set of randomly chosen figures. Our analysis revealed that most of the accurately recognized text had confidence scores between 85% and 90%. As a result, we set the threshold for confidence scores at 80% to ensure high-quality text detections. Figure 2 shows the final results of text extraction.

Incorporating this double-pass algorithm into our text detection process enables us to increase the overall accuracy and reliability of text extraction from vertical bar charts. This improvement, in turn, allows for better chart comprehension and information extraction.

Algorithm 1 Double-pass Algorithm

1: Run AWS Rekognition on the figure to detect text and obtain corresponding locations of the text
2: Select the detected text with Confidence ≥ 80
3: Compute the bounding-boxes using location output for the text obtained after Step 2
4: Fill the bounding-boxes corresponding to the selected text with white color
5: Repeat Steps 1, 2, and 3 (2nd pass)

B. Axes Detection

Most of previous methods of axes use traditional methods like Algorithm 2 [5], max consecutive black pixels to detect x-axis and y-axis seperately. But the results are not possitively satistying, which only has an accuracy of 70%. In fact, we don't need to detect the axes seperately. We can detect the bounding boxes of the plot and use the edges as axes. So we fine-tuned YOLO(You only look once)v5 model to get the bounding boxes of the plots.

C. Texts Classification

For texts classification, we made a series of rules based on the plot bounding box. First, the texts at the very bottom and on the far left are the title of the x-axis and y-axis. Second, the texts above the top edge of the plot bounding box are classified as the chart title. Third, the texts on the left of the left edge are the y-axis values. And the texts below the bottom edge are the x-axis values.

Algorithm 2 Axes Detection using traditional Method

1: Convert input image as binary image.
2: Scan the matrix vertically and trace the continuity of black pixels within adjacent columns.
3: The first column that has the highest number of continuous black pixels was identified as the y-axis.
4: To find the x-axis, use horizontal scan in step 2.

D. Bar Detection

Bar detection is a key step to data inference. Like in Step B, we also applied YOLOv5 for bar detection. However, we don't have the bounding boxes of the bars in the training data. So we constructed our fine-tuning data by manually labeled the bars bounding boxes of 300 images. Then we trained the pre-trained YOLOv5 model for 100 epochs. Figure 3 gives the example of bounding boxes of plot and bars.

E. Data Inference

With the bounding boxes of y-axis values, bars and the plot, we can calculate the actual value of the data points with:

$$e_n = (\frac{\triangle h}{h_n} + 1) \times y_n$$

Here $\triangle h$ denotes the height difference between the bar and the nth y-axis value. h_n is the height of the nth y-axis value. y_n denotes the numeric value of the nth y-axis value. We calculate the estimated value of the data point with every y-axis value. After removing the

outliers, we compute the mean of all the estimated values as the result.

V. RESULTS CONCLUSION

A. Experiment Results

In the axes and bars detection steps, the bounding box with the highest probability is chosen as the bounding box. The R^2 scores of $x_0, y_0, height, width$ are shown below. The reults here are further used to classify the texts in later steps.

	$R^2 score$
x_0	0.99730
y_0	0.99415
width	0.98352
height	0.98776

TABLE I. The R^2 scores of the bounding box.

Table II summarizes the accuracies of texts classification for entire dataset. We define accuracy as the percentage of data points from the dataset of interest that are correctly detected.

The rather low accuracy on Y-axis value is caused by the inaccuracy of OCR detection. Y-axis values are more likely to be misindentified a single text in some figures.

As for the data inference step, we achieved an overall error rate of 13.37%. Further more, we conducted our experiments on the figures whose y-axis values are correctly identified and got an average error rate of 2.32%.

B. Limitations and Future Work

First, the accuracy of the numeric value of the data points relies heavily on the results of OCR algorithm. A possible way to further improve the precision of the results is to fine tune the OCR model to better fit our task. Second, text classification algorithm based on a series of rules requires the content distribution of charts to be the same or alike. Third, to extend our model to other types of charts, one key problem is to figure out how to locate the data points in the figure. The remaining parts of our model are easily transferable and has a rather satisfying performence.

Parameter	Accuracy
Chart Title	98.95%
X-axis value	97.53%
Y-axis value	88.34%
X-axis title	99.67%
Y-axis title	98.33%

TABLE II. Accuracies of texts classification

C. Conclusion

We developed a framework implementing a set of algorithms that automatically parse numeric values and associated semantic information from bar-plots in stem textbooks. The overall performance for parsing data from a general bar-plot has been improved compared with the state-of-the-art with realworld datasets.

[1] Jinglun Gao, Yin Zhou, and Kenneth E. Barner. *View: Visual information extraction widget for improving chart images accessibility*. In 19th IEEE International Conference on Image Processing, pages 2865–2868, 2012.

[2] Manolis Savva, Nicholas Kong, Arti Chhajta, Li Fei-Fei, Maneesh Agrawala, and Jeffrey Heer. *Revision: Automated classification, analysis and redesign of chart images*. In Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, pages 393–402, 2011.

[3] Abhijit Balaji, Thuvaarakkesh Ramanathan, and Venkateshwarlu Sonathi. *Chart-text: A fully automated chart image descriptor*. arXiv preprint arXiv:1812.10636, 2018.

[4] Junyu Luo, Zekun Li, Jinpeng Wang, and Chin-Yew Lin. *Chartocr: Data extraction from charts images via a deep hybrid framework*. In 2021 IEEE Winter Conference on Applications of Computer Vision (WACV). The Computer Vision Foundation, January 2021.

[5] C. Rane, S. M. Subramanya, D. S. Endluri, J. Wu, and C. L. Giles, *ChartReader: Automatic Parsing of Bar-Plots*, 2021 IEEE 22nd International Conference on Information Reuse and Integration for Data Science (IRI), Las Vegas, NV, USA, 2021, pp. 318-325, doi: 10.1109/IRI51335.2021.00050.

[6] S. V. P, M. Yusuf Hassan, and M. Singh, *LineEX: Data Extraction from Scientific Line Charts*, 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2023, pp. 6202-6210, doi: 10.1109/WACV56688.2023.00615.

[7] P. Siekman, E. C. Botha, and L. Feng, *Visual extraction of numerical data from scatter plots*, 2021 IEEE 22nd International Conference on Information Reuse and Integration for Data Science (IRI), Las Vegas, NV, USA, 2021, pp. 326-333, doi: 10.1109/IRI51335.2021.00051.

[8] N. Yokokura and T. Watanabe, *Layout-based approach for extracting constructive elements of bar-charts*, International Workshop on Graphics Recognition, pp. 163–174, 1997.

[9] Y. Zhou, P. Yan, and L. T. Chew, *Hough technique for bar charts detection and recognition in document images*, International Conference on Image Processing (Cat. No. 00CH37101), vol. 2, pp. 605–608, 2000.

[10] Yanping Zhou, Chew Lim Tan. *Chart analysis and recognition in document images*. In: Proceedings of the Sixth International Conference on Document Analysis and Recognition, pp. 1055-1058, September 2001.

[11] K. Davila, B. U. Kota, S. Setlur, V. Govindaraju, C. Tensmeyer, S. Shekhar, R. Chaudhry. *ICDAR 2019 Competition on Harvesting Raw Tables from Infographics (Chart-Infographics)*. International Conference on Document Analysis and Recognition, pp. 1594-1599, 2019.