ChartAccessMobile: An Intelligent System for Accessible Chart Navigation on Mobile Applications

Xinjun Zhu
zxjstu@zju.edu.cn
College of Computer Science,
Zhejiang University
Hangzhou, China

Ming Shen
shenming2023@zju.edu.cn
School of Software Technology,
Zhejiang University
Ningbo, China

Sheng Zhou*
zhousheng_zju@zju.edu.cn
School of Software Technology,
Zhejiang University
Ningbo, China

Ruotian Pang
ruotianpang@zju.edu.cn
School of Software Technology,
Zhejiang University
Ningbo, China

Liangcheng Li
liangcheng_li@zju.edu.cn
College of Computer Science,
Zhejiang University
Hangzhou, China

Jiajun Bu*
bjj@zju.edu.cn
College of Computer Science,
Zhejiang University
Hangzhou, China

Abstract

Charts are essential carriers for data communication, conveying complex semantics through visual channels. However, chart components in mobile applications remain largely inaccessible to visually impaired users, often suffering from missing focus or inadequate descriptions. Current solutions face critical limitations: accessibility tools typically rely on underlying data access which is restricted in compiled applications; automated chart understanding techniques struggle with domain shift when transferred to complex mobile interfaces; and existing interaction designs force users into a trade-off between usability and functionality. To address these challenges, we present ChartAccessMobile, an intelligent system that enables accessible chart navigation through pixel-level analysis, eliminating the need for underlying data access. Informed by a formative study with 10 visually impaired users, we identified a critical need for balancing high-level summaries with on-demand data details and greater interaction autonomy. Accordingly, we constructed dedicated mobile-specific datasets and benchmark models to power a vision-based pipeline. ChartAccessMobile provides a novel multi-granularity navigation mode via an accessibility overlay, allowing users to switch between concise summaries and detailed data exploration. User evaluation demonstrates that our system significantly outperforms traditional screen readers in description quality, interaction efficiency, and overall user satisfaction.

CCS Concepts

• **Human-centered computing → Accessibility technologies**; *Accessibility design and evaluation methods*; *Mobile devices*; • **Computing methodologies** → Natural language processing.

*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
W4A'26, Dubai, United Arab Emirates
© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Keywords

chart accessibility, mobile applications, blind and visually impaired, screen readers, chart understanding

ACM Reference Format:
Xinjun Zhu, Ming Shen, Sheng Zhou, Ruotian Pang, Liangcheng Li, and Jiajun Bu. 2026. ChartAccessMobile: An Intelligent System for Accessible Chart Navigation on Mobile Applications. In *Proceedings of The 23rd International Web for All Conference (W4A'26).* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Charts, as essential carriers for data visualization, play a pivotal role in presenting complex data and revealing underlying patterns through their information-rich and intuitive visual representations [56]. However, accessing these information-dense components poses significant challenges for users with visual impairments. On mobile platforms, visually impaired users primarily rely on screen readers—such as TalkBack [22] on Android and VoiceOver [5] on iOS—to access chart content, which depend heavily on developers to provide accurate alternative text[20]. Nevertheless, charts are inherently information-dense, with their core semantics heavily dependent on visual channels. Simple chart overviews or verbatim text announcements fail to convey complete chart semantics, while crafting high-quality alternative text is both time-consuming and labor-intensive, requiring developers to possess accessibility awareness and invest additional effort. Consequently, such practices are often neglected or compromised in real-world development. Joshi et al. [28] collected 151 charts from 149 mobile applications and found that 88.1% were completely skipped by screen readers, while the remaining 11.9% provided only extremely limited readable information. These findings reveal that chart components in current mobile applications widely suffer from missing focus and inadequate descriptions, preventing visually impaired users from effectively accessing and comprehending chart information. In this work, we focus on common *static* charts in mobile apps, including bar charts, line charts, and pie charts.

Current approaches to mobile chart accessibility exhibit critical limitations that hinder their deployment in real-world scenarios. First, **the "black box" nature of mobile applications restricts third-party tools**. State-of-the-art accessibility tools [62] often rely on direct access to the DOM or underlying data structures (e.g., SVG paths). However, charts in commercial apps are frequently rendered as pixel-based bitmaps or drawn on Canvases with encapsulated data, rendering these data-dependent approaches ineffective for the vast majority of apps. Second, **the significant domain shift with other platforms**. While pixel-level parsing has matured for web and document images [1, 19, 41], these models fail to accommodate the unique constraints of mobile environments—such as small-scale rendering, embedding within complex UI hierarchies, and stylistic diversity[30]. The lack of datasets dedicated to mobile chart detection and chart-to-table conversion further impedes the transfer of these technologies. Finally, **the trade-off between usability and functionality**. Specialized assistive systems often introduce complex, non-standard gestures with steep learning curves, making them difficult to integrate into daily workflows. Meanwhile, traditional screen readers rely on rigid, linear traversal [50], compelling users to passively process redundant information without the agency to selectively access high-level summaries or specific data details.

To tackle the above challenges, in this paper, we first conduct an interview study with 10 blind screen reader users to investigate their expectations regarding mobile chart descriptions and interaction modalities. The findings revealed that participants expected chart descriptions to balance summaries with specific element details in structure, focus on key practical information in content, and employ concise and natural language in expression. Regarding interaction, participants desired greater autonomous control and expressed approval of the multi-granularity navigation mode proposed in this work. Guided by these insights, we applied automated chart understanding techniques to the mobile domain by constructing *MobileChart* (a chart detection dataset) and *CNChartTable* (a Chinese chart-to-table dataset), establishing robust baselines for mobile contexts.

Building upon the mobile chart understanding technology, we designed and implemented ChartAccessMobile, a system that performs real-time chart detection, parsing, and description generation as users navigate mobile pages. Unlike previous methods, our system operates as an accessibility service overlay that analyzes the visual UI directly, making it compatible with any application regardless of its underlying implementation. We introduce a multi-granularity navigation mode that allows users to seamlessly switch between chart summaries and element-level exploration, effectively addressing the issues of information overload and lack of agency. User study results demonstrate that ChartAccessMobile outperforms the TalkBack screen reader across all three dimensions—chart description, interaction modality, and overall experience—validating the effectiveness and practicality of the proposed system.

In summary, this paper makes the following contributions:

- An interview study with 10 visually impaired users, revealing the interaction challenges and requirements they face when accessing charts on mobile devices.
- The application of automated chart understanding techniques to the mobile domain, including the construction of a dedicated dataset and benchmark models.
- The design, development, and evaluation of ChartAccessMobile—a novel assistive system that enables visually impaired users to efficiently access charts in mobile applications.

2 Related Works

We review related technologies that assist visually impaired users in accessing charts, upon which our work builds.

2.1 Mobile Chart Access Assistive Technologies

Early work explored non-visual chart comprehension through multimodal feedback: Abu Doush et al. [18] combined haptic and audio cues with exploratory and guided interaction strategies for desktop charts. However, research on mobile platforms remains very limited. Joshi et al.[28] approached the problem from a developer perspective, proposing the CAM module for the MPAndroidChart library, which generates chart descriptions based on custom templates and provides description and focus support for chart components through accessibility interfaces; however, this method is only applicable to a specific chart library with limited generalizability, and the chart descriptions are generated from fixed templates without fully considering the cognitive needs of visually impaired users. Prakash et al.[45] provided chart magnification functionality for low-vision users, but this approach only addresses the needs of low-vision users and cannot meet the access requirements of users who are completely blind. The representative work in mobile chart accessibility is ChartA11y proposed by Zhang et al.[62], which renders chart data as SVG charts through the Chart Reader[52] visualization engine and loads them into iOS applications, providing visually impaired users with two interaction modes: semantic navigation and touch mapping. Semantic navigation allows users to browse chart elements hierarchically; touch mapping maps chart spatial layouts to the touchscreen, enabling users to explore data points through dwell and touch operations and receive feedback via speech or vibration. However, interaction methods such as touch mapping are relatively complex for visually impaired users accustomed to standard screen reader gestures, presenting high learning costs; meanwhile, ChartA11y conveys semantics based on known chart data, whereas mobile chart data is often encapsulated within applications and inaccessible to third parties, and this approach lacks chart semantic extraction capabilities and cannot be applied to charts in real mobile applications. Therefore, there remains a lack of practically deployable mobile chart accessibility systems, and we propose ChartAccessMobile to fill this gap.

2.2 Mobile Interaction with Screen Readers

On mobile platforms, visually impaired users rely on screen readers to access application content, with TalkBack on Android and VoiceOver on iOS being the most widely adopted assistive technologies. Screen readers transform two-dimensional graphical interfaces into linearized, one-dimensional sequences of screen elements for auditory navigation. Central to this interaction paradigm is the accessibility description, which developers configure through component-level accessibility properties. For each focusable interface element, the screen reader generates spoken feedback based on the description text, supplemented by information about the component's type and state. When accessibility descriptions are missing or improperly configured, screen readers may fail to focus on components entirely, or provide only uninformative feedback upon focus.

Several established guidelines address mobile accessibility requirements. The WCAG principles of Perceivability and Operability [8] specify standards for focus behavior and accessibility descriptions of non-text and non-decorative components. At the implementation level, official accessibility documentation for both Android [15] and iOS [4] platforms provides dedicated guidance on configuring component accessibility properties to ensure compliance with these standards.

Despite the availability of such guidelines and documentation, chart components present unique accessibility challenges in practice. As primary vehicles for information visualization, charts convey data distributions, temporal trends, and relational patterns through visual encodings including spatial layout, geometric forms, and color mappings—rendering their semantics heavily dependent on visual perception. This characteristic makes charts fundamentally different from conventional interface elements that can be adequately described through brief labels or summaries.

Merely announcing textual elements within a chart fragments the holistic semantics into isolated pieces lacking contextual coherence. Crafting high-quality alternative text that accurately conveys data insights demands both accessibility awareness and data interpretation skills from developers, requiring substantial time and effort. Under the constraints of project resources and deadlines, chart accessibility is frequently neglected or compromised, resulting in widespread issues where charts in mobile applications are either non-focusable or accompanied by overly brief descriptions.

The inherent linear traversal mode of screen readers further compounds the challenges of chart access. When encountering charts as high-density visualizations, existing applications typically adopt one of two strategies: presenting the chart as a single focusable element with an extended description, or decomposing it into multiple elements for sequential announcement. The former approach results in information overload, forcing users to passively receive substantial content that may not address their specific needs. The latter, when dealing with numerous data points, requires users to navigate element by element through repeated swipe gestures, precluding on-demand access to desired information and effectively diminishing user agency. To address these limitations, we conducted user studies to investigate visually impaired users' preferences for chart descriptions and propose a multi-granularity navigation interaction model.

2.3 Automatic chart understanding technology

Automated chart understanding techniques aim to perceive charts and parse their visual and semantic elements, transforming chart content into machine-operable structured representations to support downstream applications such as chart question answering and summary generation, ultimately helping users access and comprehend the information conveyed by charts [21, 25]. This technology encompasses subtasks including chart detection, chart-to-table conversion, and chart description generation, with several studies applying it to chart accessibility in web and document contexts [1, 19, 41].

Chart detection serves as the initial stage of chart understanding, with the core objective of locating chart regions within pages to provide input for subsequent processing. Existing work primarily targets PDF documents and web pages. PDFFigures 2.0 [14] extracts figures from academic papers by analyzing document structure and caption-figure layout relationships. GraphDoc [12] introduces layout relationship modeling to achieve chart region extraction and caption pairing. PixelWeb [61] combines web rendering results with DOM structure for element localization. These methods implicitly assume that charts possess clear boundaries and standardized layouts. However, mobile charts are embedded as UI components within applications featuring complex layouts, and their structures are prone to degradation due to small-scale rendering. Furthermore, view hierarchy information is difficult for third parties to access, resulting in significant domain shift [17] that prevents effective transfer of existing methods. Currently, chart detection for mobile platforms remains underexplored, lacking publicly available benchmark datasets and models.

Chart-to-table conversion aims to extract underlying data from chart images and construct structured tables, providing a data foundation for downstream tasks. DePlot [31] formulates this task as an image-to-text translation problem, enabling end-to-end conversion. UniChart [39] learns relationships between graphical primitives and data through domain-specific pretraining. OneChart [10] achieves state-of-the-art performance with a lightweight model while introducing reliability verification mechanisms. However, these methods are all built upon English chart datasets such as ChartQA [38] and PlotQA [40], facing challenges in semantic alignment and insufficient vocabulary coverage when processing Chinese charts, rendering them unable to effectively generate Chinese content. Similarly, publicly available benchmark datasets and models for Chinese chart-to-table conversion are currently lacking to support chart understanding research and applications in Chinese contexts.

Chart description generation aims to convert key information from charts into natural language to help users understand chart content. Obeid and Hoque [43] proposed early Chart-to-Text work and constructed the first large-scale chart summarization dataset. VisText [51] introduced multi-level semantic annotations including statistical relationships and trends. MatCha [32] enhanced end-to-end generation capabilities through pretraining tasks such as chart derendering. This task depends on outputs from upstream stages, yet the current lack of benchmarks for mobile chart detection and Chinese chart-to-table conversion leaves description generation without effective support, hindering high-quality chart information delivery for visually impaired users. In summary, chart understanding techniques for Chinese mobile contexts remain insufficiently studied, lacking publicly available benchmark datasets and models. Therefore, we construct datasets and propose baseline models, generating chart descriptions that align with the cognitive needs of visually impaired mobile users based on user study findings, thereby applying automated chart understanding techniques to mobile chart accessibility.

3 Formative Study

To deeply understand the current status and preferences of screen reader users regarding chart descriptions and interaction methods in mobile chart access scenarios, we conducted a user needs study through semi-structured interviews with 10 screen reader users.

3.1 Participants

We recruited 10 visually impaired participants, aged between 27 and 50, comprising 6 totally blind and 4 with low vision. Their diverse backgrounds included 2 computer-related practitioners, 7 non-computer workers, and 1 accessibility consultant. Five were relatively familiar with charts; five were not. All were proficient screen reader users and frequent internet users.

3.2 Procedure

This study employed face-to-face semi-structured interviews, following a unified process while allowing participants to freely express their views. The interviews focused on three core questions:

Q1: Are visually impaired users satisfied with the accessibility of charts in current mobile applications?

Q2: What kind of chart descriptions do visually impaired users expect?

Q3: What interaction methods do visually impaired users prefer when interacting with charts in mobile applications?

For Q1, we provided representative chart cases from current mobile applications for users to experience and evaluated them from three dimensions—perceivability, understandability, and operability—referencing the core principles of WCAG accessibility guidelines [8].

For Q2, based on W3C WAI guidelines for complex images [58], which propose that complex images require two parts of alternative text (a brief description for quick identification and a long description for presenting key information), and Diagram Center's recommendation for layered expression of charts [9], we designed chart descriptions comprising chart summaries and chart element information. Chart summaries were structured according to Lundgard et al.'s four-level semantic content model [34]: basic information includes chart title and type to help users establish overall cognition; statistical information covers key metrics such as mean, extrema, and range to characterize numerical features and comparative relationships; cognitive information describes perceptible phenomena requiring inference, such as trend patterns in time-series charts, proportion distributions in compositional charts, and category differences in comparative charts; auxiliary background information conveys social and contextual interpretations that depend on readers' domain-specific knowledge. Chart element information includes title, data points, axis labels, legend, component colors, and axis scale ranges. Participants evaluated the necessity of both parts and the importance of each component. Additionally, we designed three description examples with different language styles to explore user preferences: the *original description* refers to the chart description provided by the original application; the *detailed description* covers complete chart summaries and all chart elements, offering comprehensive but relatively verbose content; and the *concise description* removes redundant and non-essential information from the detailed description while optimizing language expression, resulting in a more natural, succinct, and key-focused description.

For Q3, to address the limitations of screen readers' linear traversal mode—namely information redundancy and lack of user agency—we designed a multi-granularity navigation interaction mode. This mode introduces a hierarchical access mechanism built upon conventional single-finger left-right swipe gestures: when focus moves to a chart region, the system first provides a voice prompt asking whether to announce the chart summary and informs users that a two-finger swipe can access chart elements; users can double-tap to listen to the chart summary, or use a two-finger swipe to enter element-by-element access, navigating between different elements via two-finger left-right swipes; if users are not interested in the current chart or have obtained the desired information, they can skip the chart directly with a single-finger swipe. Through this design, users can autonomously select the access granularity according to their actual needs, enabling different levels of chart access including quick summary overview, element-by-element exploration, or direct skip. Participants accessed the same chart using both the screen reader and the multi-granularity navigation mode, then provided evaluations with explanations.

After interviews, all audio materials were transcribed verbatim to form complete interview documents. Based on this, we employed thematic analysis [6] to systematically summarize text content, extracting users' focal points and core needs in chart access processes.

3.3 Findings

Inadequate chart accessibility.Almost all participants expressed dissatisfaction with the accessibility of charts in current mobile applications. Regarding perceivability, most participants reported difficulty perceiving the existence of charts through screen readers, often misidentifying them as ordinary text information. P3 mentioned: *"The speech just reads me a string of numbers; I have no way of realizing this is a chart."* P7 expressed similar confusion: *"Sometimes I swipe past without knowing it's a chart; I thought it was just a few lines of text."* Regarding understandability, participants generally believed that existing descriptions fail to convey complete and logically clear chart semantics. P5 explicitly stated: *"The information I get is fragmented and partial, without an overall explanation."* Regarding operability, participants found the linear traversal mode of screen readers cumbersome. P2 complained: *"I have to swipe one by one; it's especially troublesome when there are many data points."* P9 added: *"Too many swipes—by the end, I've forgotten what was said at the beginning."*

Preferences for chart description content and style. Regarding language style, most participants clearly stated a preference for concise descriptions rather than lengthy detailed content. P1 explained: *"I don't need to hear that much; just tell me what's most important."* P4 shared the same view: *"Long descriptions are tiring to listen to, and it's easy to lose focus, making it harder to remember the key points."* Additionally, multiple participants suggested reducing the use of technical terminology. P8 stated: *"Terms like axes, legends—not everyone understands them. Can you use more straightforward language?"*

Regarding description structure, almost all participants considered both chart summaries and chart element descriptions indispensable. P2 pointed out: *"The summary lets me know what this chart is roughly about, but I also want to know the specific data."* P10 added: *"Both are needed—first hear the general idea, then listen to details if I want to learn more."*

Regarding chart element information, participants generally considered data points and titles the most core content. P3 emphasized: *"Data points are the most important; isn't a chart about the data?"* However, for axis labels and legends, some participants expressed difficulty understanding due to lack of relevant knowledge. P7 admitted: *"I'm not very clear about what a legend is; I don't deal with charts much."* Regarding component colors and scale ranges, participants unanimously agreed these need not be provided. P5 directly stated: *"Colors have no meaning for me; no need to mention them."*

For summary content, participants valued information that helps quickly establish chart cognition but showed little interest in auxiliary background information. P8 stated: *"Background information wouldn't be useful to me even if explained."*

In summary, visually impaired users prefer to receive key information with practical significance, consistent with their preference for concise, non-redundant language styles.

Interaction should enhance user agency. Participants' main dissatisfaction with screen readers' linear traversal mode was the lack of selectivity. P1 stated: *"Right now I'm passively listening one by one; I can't even skip some unimportant parts."* P4 also mentioned: *"Sometimes I just want to know the highest and lowest points, but I have to listen to all the data."*

In contrast, most participants expressed approval of the multi-granularity navigation mode proposed in this study. P6 commented: *"This approach is better; I can choose to listen to the summary first, then continue if I want details."* P9 believed this mode *"provides more selectivity, making it easier to get the information I want."*

Summary. Participants identified multiple issues with current mobile application chart access, including difficulty perceiving chart existence, fragmented and incomplete description information, and the cumbersome linear traversal mode lacking user agency. To address these issues, participants expect chart descriptions to balance summaries with specific element descriptions in structure, focus on key practical information in content, and employ concise and natural expression in language. Regarding interaction, participants desire stronger autonomous selection capabilities, with the multi-granularity navigation interaction mode being endorsed by users. Based on these findings, we designed and developed the ChartAccessMobile system, detailed in the following section.

4 ChartAccessMobile

Drawing on insights from the formative study and aligning with our system design goals, we designed ChartAccessMobile, an intelligent system that makes mobile application charts accessible. The system can be deployed as an Android accessibility service, enabling real-time interpretation of charts across any mobile application without requiring modifications to the original apps.

Unlike existing mobile chart accessibility tools that rely on underlying chart data, ChartAccessMobile directly utilizes vision algorithms to analyze chart image pixels and extract semantic information, capable of handling charts rendered as bitmaps or canvases in commercial applications—effectively penetrating the "black box" that third-party assistive tools previously could not access. Combined with the semantic reasoning capabilities of large language models, ChartAccessMobile serves as a universally deployable solution for mobile chart accessibility.

The overall pipeline is illustrated in Figure 1. ChartAccessMobile comprises three chart understanding modules and one interaction module:

(1) **Chart Detection Module:** Localizes chart regions and identifies chart types from mobile application pages. We constructed the MobileChart dataset and selected YOLOv11-n as the baseline model, achieving real-time detection performance suitable for mobile deployment.
(2) **Chart-to-Table Module:** Converts chart images into structured data tables, extracting values, axis labels, legends, and other information. We constructed the CNChartTable dataset and proposed the DePlot-Zh model, supporting both Chinese and English chart parsing.
(3) **Chart Description Generation Module:** Generates chart descriptions comprising chart summaries and data point descriptions based on structured data and user study findings. Using Qwen3-8B with Chain-of-Thought prompting, it produces concise, accurate, and naturally fluent descriptions.
(4) **System Design and Implementation:** Provides a multi-granularity navigation interaction mode with an accessible overlay interface. Users can access chart summaries or data points on demand through gestures, with support for voice interaction, repeat playback, and auto-playback functions to enhance user agency.

4.1 Chart Detection

The chart detection module aims to localize chart regions and identify chart types from mobile application pages, providing input for subsequent chart parsing and description generation.

Dataset Construction. Due to the lack of chart detection datasets dedicated to mobile scenarios, we constructed the MobileChart dataset using a hybrid strategy combining real and synthetic samples. For real samples, we crawled 300 mobile applications from Google Play and Wandoujia app markets, covering chart-frequent categories such as health & fitness, personal finance, and weather. After manual screening and annotation, we obtained 273 chart-containing page samples. However, the proportion of mobile pages containing charts is relatively low, and such pages are typically located in deep application paths, making manual collection insufficient for large-scale dataset construction. Therefore, we proposed an automated data synthesis method comprising three core steps: (1) collecting mobile application screenshots and layout information via Appium-based[46] automated crawlers; (2) filtering regions suitable for chart embedding from page layouts based on heuristic rules, including navigation bar filtering, geometric constraint filtering, and component semantic filtering to ensure embedding rationality; (3) embedding charts into page slots using a contain scaling mode and applying context-aware color filling for margin regions to achieve natural visual integration. The resulting MobileChart dataset contains approximately 40k samples.

Model Selection. Based on the MobileChart dataset, we conducted comparative experiments on multiple mainstream object detection models, including YOLOv12 [55], YOLOv11 [54], and YOLOv8 [53] series at different scales, as well as classic SSD [33] and Faster R-CNN [49] models, evaluating them across dimensions of detection accuracy, inference efficiency, and model size. The

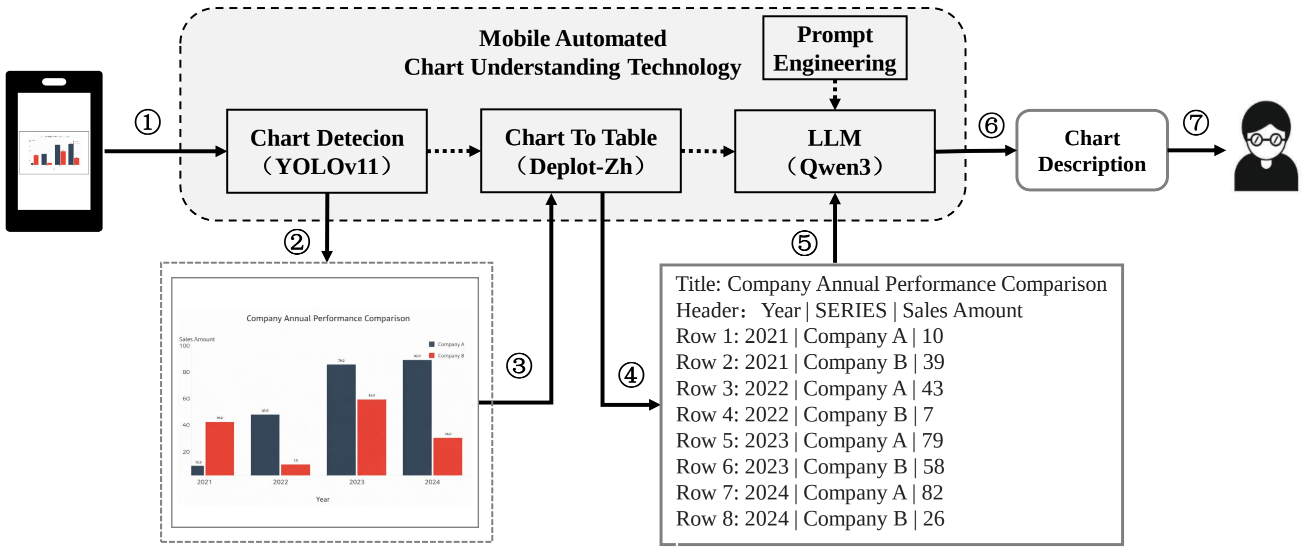


Figure 1: Mobile chart understanding pipeline: YOLOv11 for chart detection, DePlot-Zh for chart-to-table conversion, and Qwen3 with prompt engineering for description generation.

experimental results are shown in Table 1. In terms of detection accuracy, the YOLO series models outperform traditional detectors overall, with YOLOv11 series achieving the highest accuracy. Regarding inference efficiency, the latency of YOLO n-scale models meets mobile real-time response requirements, while Faster R-CNN exhibits significantly higher latency due to its two-stage architecture. For deployment feasibility, n-scale models with parameters below 4M and computational cost below 9 GFLOPs satisfy mobile lightweight deployment requirements. Considering all factors, we selected YOLOv11-n as the benchmark model for mobile chart detection, achieving a test set mAP@[0.5:0.95] of 85.10 with inference latency of 27.2ms, representing the optimal balance among detection accuracy, response speed, and deployment cost.

Table 1: Performance comparison of detection models on the MobileChart dataset.

Model	mAP@[0.5:0.95] (%) ↑	Latency (ms) ↓	Params (M) ↓	FLOPs (G) ↓
YOLOv12-s	85.63	31.4	9.29	21.69
YOLOv12-n	84.12	30.8	**2.60**	6.65
YOLOv11-s	**88.24**	34.7	9.46	21.72
YOLOv11-n	85.10	27.2	2.62	6.61
YOLOv8-s	84.50	35.9	11.17	28.82
YOLOv8-n	83.40	**14.2**	3.16	8.86
Faster R-CNN	80.62	167.5	41.76	268.76
SSD	73.09	33.0	3.44	**1.24**

4.2 Chart-to-Table

The chart-to-table module aims to convert chart images into structured data tables, providing data support for downstream description generation tasks and serving as a core component in the chart semantic extraction pipeline.

Dataset Construction. Existing chart-to-table datasets are almost exclusively designed for English scenarios, with insufficient support for Chinese. To address this, we constructed the Chinese chart-to-table dataset CNChartTable using a programmatic chart generation paradigm, comprising three stages: (1) batch collection of high-quality structured data tables from the Kaggle [29] platform; (2) automatic parsing of tabular data using the DeepEye[36, 37] visualization recommendation system to generate chart metadata, followed by Chinese semantic conversion via large language models; (3) design of three code templates—MPAndroidChart[27], ECharts [3], and Matplotlib [26]—targeting mainstream implementation approaches for mobile charts, with random sampling of element and style parameters to generate chart images that closely resemble real-world scenarios with diverse visual styles. Additionally, we collected charts from real mobile applications and manually annotated their underlying data tables as real samples. The resulting CNChartTable dataset contains approximately 47k synthetic samples and 273 real samples, covering three chart types: line, bar, and pie charts.

Model Design. Building upon the DePlot model, we propose DePlot-Zh, which achieves effective adaptation to the Chinese chart-to-table task through three key designs: (1) Chinese vocabulary expansion: adopting an incremental vocabulary extension strategy to incorporate high-frequency Chinese characters and words from general Chinese corpora into the existing vocabulary, improving the completeness and decoding efficiency of Chinese text generation; (2) linearized table design: supplementing the original DePlot output with semantic fields such as y-axis labels to ensure the output structure covers complete chart semantic information; (3) parameter-efficient fine-tuning: introducing the LoRA method to achieve lightweight incremental adaptation while preserving pretrained capabilities, with gradient updates enabled for the decoder's embedding layer and output projection layer to accommodate the expanded Chinese vocabulary.

Prompt for Chart Desription Generation

Task Description: You are a chart accessibility expert. Generate concise, accurate, and natural chart descriptions for BVI users.

Input Format: Structured chart data containing TITLE, TYPE, dimension labels, data points, and statistics (MAX, MIN, AVG, DIFF).

First, parse the structured input to extract chart information and establish semantic understanding of the chart.

Second, generate **BASIC INFO** from TYPE and TITLE fields. If TITLE is missing, infer the theme from data content and dimension labels.

Third, generate **STATISTICAL INFO** by selecting 2-4 key metrics (from MAX, MIN, AVG, DIFF) that best reflect data patterns. Use natural language; avoid technical jargon.

Fourth, generate **COGNITIVE INFO** based on chart type:
- Time-series charts: Describe overall trends and significant fluctuations
- Composition charts: Describe proportion distribution characteristics
- Comparison charts: Highlight differences between categories

Fifth, compose **CHART SUMMARY** by integrating basic, statistical, and cognitive information into one coherent paragraph.

Sixth, construct **DATA POINT TEMPLATE** with placeholders ({x_value}, {y_value}, {series} if multi-series). Validate by filling with real data points.

Output Format:
Chart Summary: (one natural paragraph)
Data Point Template: (one template sentence)

Example 1 (Bar Chart):
Input: TITLE: US Annual Inflation Rate (2020-2024), TYPE: Bar Chart, Year | Rate (%), Data: 2020|1.2, 2021|4.7, 2022|8.0, 2023|4.1, 2024|2.9, MAX:8.0, MIN:1.2, AVG:4.2
Output:
Summary: This bar chart shows US annual inflation rates from 2020 to 2024. The average rate is approximately 4%, peaking at 8.0% in 2022 and dropping to 1.2% in 2020. Inflation rose steadily from 2020, peaked in 2022, then declined to around 3% by 2024.
Template: The inflation rate in {x_value} was {y_value}%.

Example 2 (Multi-series Line Chart):
Input: TITLE: Tokyo vs Sydney Temperature Comparison, TYPE: Line Chart, Month | Temp (°C) | SERIES, Data: Jan|5.4|Tokyo, Jul|25.9|Tokyo, Jan|22.8|Sydney, Jul|12.7|Sydney, MAX:25.9, MIN:5.4, AVG:16.9
Output:
Summary: This line chart compares average temperatures between Tokyo and Sydney. The overall average is around 17°C, with Tokyo in July being the highest at 25.9°C and Tokyo in January the lowest at 5.4°C. Tokyo shows greater seasonal variation while Sydney remains relatively stable.
Template: {series} average temperature in {x_value} is {y_value}°C.

Example 3 (Pie Chart): *Input:* 2024 Chinese Urban Household Monthly Expenditure, Pie Chart, MAX:28.5, MIN:6.2
Summary: This pie chart shows the composition of monthly expenditure for Chinese urban households in 2024. Food & tobacco accounts for the highest proportion at 28.5%, while other expenses account for the lowest at 6.2%...
Template: {x_value} accounts for {y_value}%.

Figure 2: Prompt for generating chart summaries and data point description templates using Chain-of-Thought and few-shot prompting strategies

Comparative Experiments. We conducted comparative experiments with ChartOCR [35] (an OCR-driven pipeline method), ChartReader [13] (a multi-task joint modeling method), and UniChart [39] (a general chart understanding multimodal model), evaluating model performance on three datasets: CNChartTable, ChartQA [38], and Chart2Text [43]. The experimental results are shown in Table 2. DePlot-Zh achieves optimal or near-optimal performance across all three datasets. On the Chinese dataset CNChartTable, DePlot-Zh significantly outperforms other methods due to its vocabulary expansion and model architecture advantages. On English datasets, DePlot-Zh preserves its pre-trained capabilities through parameter-efficient fine-tuning, achieving performance comparable to UniChart. These results demonstrate that DePlot-Zh possesses strong Chinese chart parsing capabilities while maintaining good performance on English charts.

Table 2: Comparative experimental results on Chart-To-Table task.

Method	CNChartTable		ChartQA		Chart2Text	
	RMS	RNSS	RMS	RNSS	RMS	RNSS
ChartOCR	20.14	28.37	42.45	63.21	34.18	55.43
ChartReader	23.26	33.15	66.38	87.42	57.53	79.16
UniChart	81.42	92.18	83.67	**94.23**	74.31	81.47
DePlot-Zh (Ours)	**85.25**	**95.63**	**84.14**	92.46	**77.32**	**82.58**

4.3 Chart Description Generation

The chart description generation module produces natural language descriptions tailored to the needs of visually impaired users based on the structured data output from the chart-to-table module.

Description Structure Design. Based on user study findings, we designed a chart description structure comprising chart summary and data point descriptions to meet visually impaired users' needs for both summary and specific element descriptions.

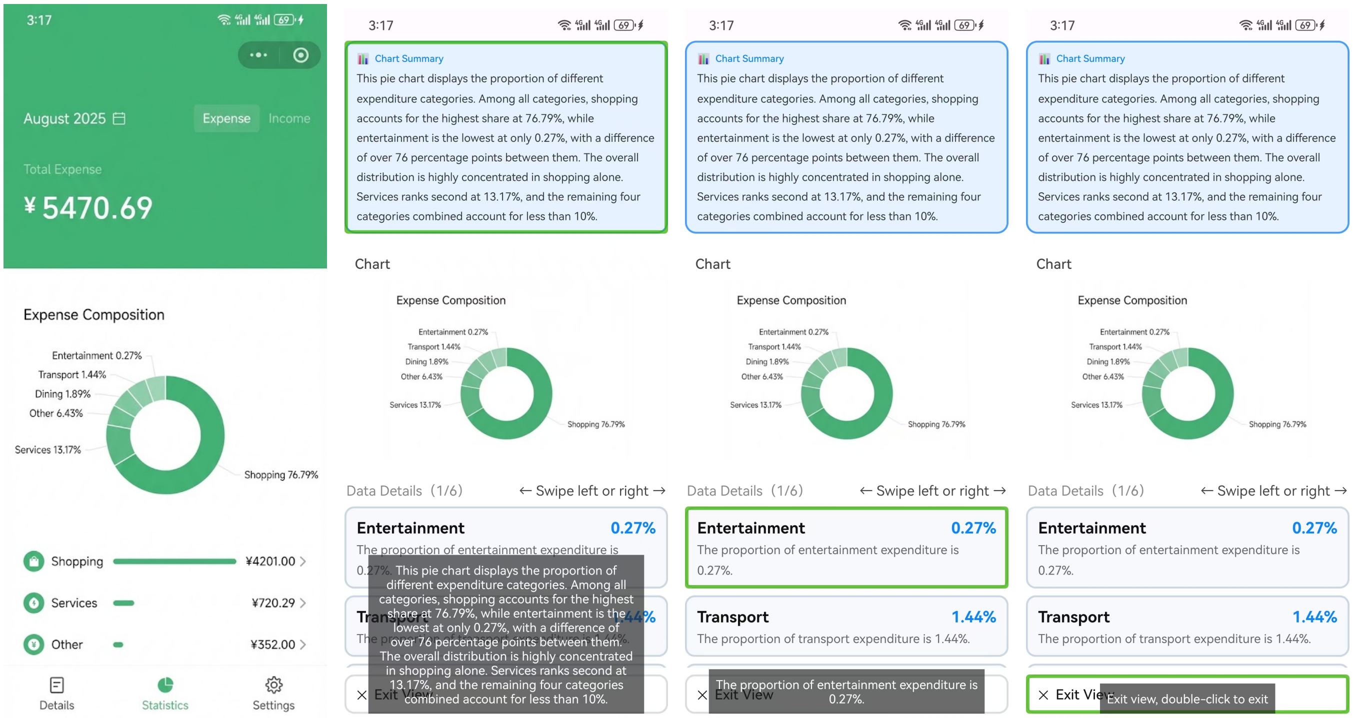
The chart summary includes three types of information: (1) basic information, including chart type and title; (2) statistical information, such as maximum, minimum, and average values; (3) cognitive information, such as data trends and significant features. Auxiliary background information was removed as participants generally considered it of limited practical value.

For chart element information, our study showed that data points and title are the core elements users care about most, but title already appears in the summary. Axis labels and legend, while unfamiliar as terms, are essential for interpreting data points, so they are integrated into data point descriptions rather than presented separately. Component colors and axis scales were removed as participants indicated no need for visual information. Thus, chart element descriptions comprise x-axis value, y-axis value, series value, and axis labels for each data point.

Based on participants' preference for concise expression in our study, chart descriptions employ concise and natural language, avoiding redundancy and technical terminology.

Generation Method. Chart summary generation requires flexible reasoning to integrate basic facts, statistical highlights, and type-varying cognitive interpretations into natural, concise language that meets user preferences identified in our formative study.We therefore implement description generation based on large language models and prompt engineering, using Qwen3-8B [60] as the base model, while data point descriptions use LLM-generated templates populated with extracted values to ensure factual accuracy. Following the prompt design framework proposed by Google Cloud [23], we structure the prompt into four core components: role definition, input-output specification, generation step guidance, and few-shot examples. The complete prompt structure with few-shot examples is illustrated in Figure 2.

The prompt first positions the model as a "chart accessibility expert" serving visually impaired users, with the generation objective of producing concise, accurate, and naturally fluent chart descriptions. Input adopts a structured text format containing chart metadata, data point sequences, and pre-computed statistical features; output is constrained to two components—chart summary



(a) Original app interface (b) Chart summary focused (c) Data point focused (d) Exit button focused

Figure 3: Accessibility overlay interface and interaction states for multi-granularity navigation.

and data point description template—with constraint rules ensuring format consistency.

For generation guidance, we employ Chain-of-Thought [59] prompting strategy to decompose the generation process into multiple reasoning steps: first parsing input data to establish semantic understanding, then sequentially generating basic information, statistical information, and cognitive information, and finally integrating them into a coherent summary. Cognitive information generation adopts type-specific strategies: describing temporal trends for time-series charts, depicting proportion characteristics for compositional charts, and characterizing magnitude relationships for comparative charts. To reduce numerical hallucination risks, we reference the Program of Thoughts [11] paradigm, pre-computing statistical indicators such as mean and extrema through programs in preprocessing for model reference, avoiding errors caused by model self-computation.

For data point descriptions, prompts guide the model to generate reusable templates containing placeholders, incorporating axis label semantics before uniformly populating for each data point to balance natural expression with factual accuracy.

The prompt also provides three few-shot [7] examples covering bar charts, line charts, and pie charts, with outputs initially generated by GPT-4o and refined by accessibility domain experts. Based on user study feedback, the prompt explicitly requires the model to adopt concise and natural language style, avoiding redundant expressions and technical terminology.

4.4 System Design and Implementation

Interactive Interface and Pipeline. Figure 3 illustrates the interface and interaction states of ChartAccessMobile. Figure 3(a) shows the original application interface containing a chart. When users access mobile application pages, ChartAccessMobile detects charts in the page through mobile automated chart understanding technology in the background, parses them, generates descriptions, and constructs an accessible overlay based on chart understanding results.

The overlay contains virtual accessibility nodes for chart summary, individual data points, and an exit button, supporting screen reader focus and description reading, as shown in Figures 3(b), (c), and (d). After overlay construction, the system announces "accessible chart available" to alert users. Users can invoke the overlay through the accessibility shortcut button or volume keys.

Upon invocation, the system defaults focus to the chart summary node and announces "double-tap to read summary, swipe right to access data points." Users can then double-tap the summary node to hear the chart summary, or use single-finger left/right swipes to switch focus within the overlay to access individual data point information. Users can also exit the overlay through volume keys, accessibility button, or exit button, providing autonomous selection capability for chart information access.

Additionally, the system provides repeat playback and auto-playback functions: when focused on a node, users can double-tap with one finger to repeat the current node description, or swipe right with two fingers to automatically play subsequent node descriptions continuously. The system also integrates voice interaction functionality; overlay invocation, navigation, exit, repeat playback, and auto-playback all support voice command triggers.

System Implementation. ChartAccessMobile is implemented based on the Android Accessibility Service framework [16], adopting an edge-cloud collaborative architecture. The chart detection model is deployed on mobile devices to ensure real-time response, while chart-to-table and description generation models are deployed on servers to balance computational requirements.

On the mobile side, the system monitors foreground application window content change events through the accessibility event mechanism. Upon receiving event callbacks, the accessibility service captures the current page image and inputs it to the local chart detection model. To optimize resource utilization and reduce server request overhead, the system implements two key mechanisms. The page monitoring mechanism triggers chart detection only when the foreground application undergoes page transitions, avoiding redundant scanning of the same page. The chart caching mechanism computes perceptual hash values [57] for parsed chart images and stores them locally; subsequently detected charts are compared using Hamming distance [24] to determine whether they have already been processed, preventing duplicate server requests. For mobile deployment, the YOLOv11-based detection model is converted from PyTorch format to NCNN [42] format, compiled via NDK, and invoked through JNI bridge layer to optimize inference efficiency on mobile devices.

On the server side, the system adopts a microservice architecture divided into a business layer and an inference layer, each deployed and managed as independent services. The business layer, built on Spring Boot [44], handles request reception, task scheduling, rate limiting, and concurrency control through blocking queues and thread pools. The inference layer, built on FastAPI [48], integrates PyTorch [47] for model loading and inference computation. To ensure high availability, the inference layer employs clustered deployment with service registration and discovery through Nacos [2], enabling load-balanced request distribution across nodes. Data transmission employs HTTPS encrypted protocols with gzip compression for text-based chart information to reduce bandwidth overhead.

Upon receiving chart descriptions from the server, the system constructs a `TYPE_ACCESSIBILITY_OVERLAY` floating window as the chart access overlay, creating virtual accessibility nodes for chart summary, individual data points, and exit button through the `ExploreByTouchHelper` interface, with corresponding accessibility descriptions configured to support screen reader focus and announcement. The overlay also displays the detected chart image to assist users with low vision.

The system supports both gesture and voice interaction modalities. Overlay invocation and dismissal are implemented through physical volume key monitoring or accessibility shortcut button clicks, avoiding occupation of screen reader gestures. Volume key interaction is achieved by monitoring key events within the accessibility service, with the service caching key sequences based on time windows and performing pattern matching to support various trigger patterns such as simultaneous press, double-tap, and long-press. Overlay navigation follows conventional screen reader operations, where users can switch focus via single-finger left/right swipes. Repeat broadcast and auto-broadcast functions are triggered through `ACTION_CLICK` events on virtual nodes and two-finger swipe gestures captured by touch event listeners on the access overlay.

Voice interaction is implemented through a lightweight wake word detection model running in the background, monitoring for preset wake phrases. This model focuses on single wake word recognition, offering lower computational overhead and higher accuracy compared to full speech recognition systems. After overlay invocation, speech recognition employs Voice Activity Detection (VAD) to determine when user input ends, and uses keyword matching to compare recognition results against preset commands. To enhance interaction efficiency, the system introduces a session window mechanism: after successful command execution, an 8-second wake-word-free session window opens, during which users can issue subsequent commands directly without repeating the wake word.

5 User Study

To evaluate the practical effectiveness of ChartAccessMobile, we invited 5 of the 10 visually impaired users who participated in the formative study for a follow-up evaluation (3 totally blind and 2 with low vision; age range 27–45). All participants had at least two years of experience using smartphones and screen readers, and were proficient in using TalkBack for daily application access.

The study employed a within-subjects design, where each participant experienced chart access in mobile applications under two conditions: TalkBack screen reader (baseline condition) and ChartAccessMobile (experimental condition). To control for learning and order effects, the presentation order of the two conditions was counterbalanced across participants using a Latin square design. The experimental materials consisted of 3 charts collected from real mobile applications, covering three common types: line charts, bar charts, and pie charts. Each participant first received approximately 5 minutes of training to familiarize themselves with ChartAccessMobile's interaction gestures and navigation logic, then accessed charts under each condition and freely explored chart content. After the experience sessions, participants rated chart description, interaction method, and overall experience on 5-point Likert scales (1 = very dissatisfied, 5 = very satisfied). Semi-structured interviews were also conducted to collect subjective feedback.

Results show that ChartAccessMobile significantly outperformed TalkBack across all evaluation dimensions. For **chart description**, ChartAccessMobile scored 4.40 (SD=0.55), significantly higher than TalkBack's 3.43 (SD=0.79). Participants gave positive evaluations of chart summaries generated by our system, considering the information comprehensive and practical, with natural and fluent expression. P3 commented: *"This description clearly explains the key points of the chart, such as what the overall trend is and where the maximum and minimum values are, unlike some screen readers that just mechanically read out text labels."* Participants also appreciated the data point description functionality, indicating that beyond understanding overall chart content, they sometimes need to know specific values to support deeper analysis. P5 noted: *"Sometimes I*

Table 3: User study results comparing ChartAccessMobile and TalkBack (5-point Likert scale, higher is better). Standard deviations are shown in parentheses.

Condition	Description	Interaction	Overall
TalkBack	3.43 (0.79)	3.85 (0.69)	3.64 (0.75)
ChartAccessMobile	**4.40** (0.55)	**4.51** (0.50)	**4.46** (0.52)

just want to know what a specific number is, and this feature meets my on-demand query needs perfectly."

For **interaction method**, ChartAccessMobile scored 4.51 (SD=0.50), significantly higher than TalkBack's 3.85 (SD=0.69). Participants generally expressed strong appreciation for the multi-granularity navigation mode, considering it to have clear advantages in flexibility and efficiency of information retrieval. P1 stated: *"I can first listen to the overall introduction to get a general idea, then dive into details if I'm interested in a particular part. This approach is much more efficient than swiping through items one by one—I feel more in control."* P4 added from a comparative perspective: *"The traditional screen reader approach is too linear—all the information is arranged in one sequence, and I often have to listen to a lot of unnecessary content before finding the information I want."* P2 further noted the cognitive advantages of hierarchical navigation: *"This layered approach helps me build a mental structure of the chart, knowing where I am now and what other content I can explore."*

For **overall experience**, ChartAccessMobile scored 4.46 (SD=0.52), significantly higher than TalkBack's 3.64 (SD=0.75). Participants unanimously agreed that ChartAccessMobile significantly improved the efficiency and satisfaction of chart access. P2 summarized: *"Before, I basically skipped charts when I encountered them on my phone, because the information I heard was too fragmented to piece together a complete picture. Now I feel like charts are actually usable."* P5 commented from the perspective of daily application scenarios: *"For things like checking exercise data, I used to rely on others to describe it for me. Now I can do it independently, and this independence is very important to us."*

During the semi-structured interviews, participants also offered several suggestions for improvement. Some users suggested adding customization options for speech rate to accommodate different listening preferences. Additionally, some users suggested providing more dimensions of data filtering and comparison functions for complex charts. These feedback provide valuable directions for future system iterations.

These results demonstrate that ChartAccessMobile can effectively improve visually impaired users' experience when accessing charts on mobile devices, receiving positive evaluations in chart description quality, interaction flexibility, and overall satisfaction, validating the effectiveness and practicality of our proposed approach.

6 Conclusion and Future Work

6.1 Conclusion

This paper investigates the accessibility barriers faced by visually impaired users when accessing charts on mobile devices. To gain in-depth understanding of user needs, we conducted an interview study with 10 visually impaired screen reader users. Our findings revealed that users expressed dissatisfaction with the perceivability, understandability, and operability of charts in current mobile applications. Users expect chart descriptions to balance overall summaries with specific data in structure, focus on key practical information in content, and employ concise and natural language in expression. Users desire greater autonomy in interaction and endorsed the multi-granularity navigation interaction mode proposed in this work. Based on these findings, we designed and implemented ChartAccessMobile, a system that applies automated chart understanding techniques to mobile chart accessibility. For mobile chart detection, we constructed the MobileChart dataset containing approximately 40k samples and selected YOLOv11-n as the benchmark model, achieving real-time detection performance suitable for mobile deployment. For Chinese chart-to-table conversion, we constructed the CNChartTable dataset containing approximately 47k samples and proposed the DePlot-Zh model supporting both Chinese and English chart parsing. The system generates chart descriptions tailored to user needs based on large language models and prompt engineering, presents chart information through an accessibility overlay, and provides a multi-granularity navigation interaction mode that enables users to access chart summaries or data point information on demand, effectively addressing the information redundancy and diminished user agency caused by the linear traversal mode of screen readers. User evaluation demonstrates that ChartAccessMobile significantly outperforms traditional screen readers across three dimensions—chart description, interaction method, and overall experience—validating the effectiveness and practicality of the proposed approach.

6.2 Future Work

Despite the positive results achieved by ChartAccessMobile, several directions remain for future improvement:

- **Chart Type Extension.** The current system supports line, bar, and pie charts. Future work will extend coverage to scatter plots, radar charts, and other types.
- **Broader Application Contexts.** Beyond native apps, charts frequently appear in search results and web content where raw source data is unavailable. Future work will extend ChartAccessMobile to iOS and in-browser contexts, broadening its reach.
- **Interaction Enhancement.** Future work will explore richer modes such as natural language question-answering, enabling users to query specific chart information on demand.

Acknowledgments

We sincerely thank all the visually impaired participants who generously contributed their time and insights to this research. We are grateful to the 10 participants in our formative study for sharing their experiences and expectations regarding mobile chart accessibility, and to the 5 participants in our user evaluation for their valuable feedback on ChartAccessMobile. Their perspectives were instrumental in shaping the design and development of this system.

This work was supported by the National Natural Science Foundation of China (Grant No.62372408).

References

[1] Md Zubair Ibne Alam, Shehnaz Islam, and Enamul Hoque. 2023. Seechart: Enabling accessible visualizations through interactive natural language interface for people with visual impairments. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*. 46–64.

[2] Alibaba. 2025. *Nacos: Dynamic Naming and Configuration Service.* https://nacos.io/

[3] Apache Software Foundation. 2025. *Apache ECharts.* https://github.com/apache/echarts

[4] Apple. 2022. *Charting data — Human Interface Guidelines.* https://developer.apple.com/design/human-interface-guidelines/charts

[5] Apple. 2025. *Use VoiceOver Recognition on your iPhone or iPad.* https://support.apple.com/en-us/111799

[6] Virginia Braun and Victoria Clarke. 2006. *Using Thematic Analysis in Psychology.* https://www.tandfonline.com/doi/abs/10.1191/1478088706qp063oa

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[8] Ben Caldwell, Michael Cooper, Loretta Guarino Reid, Gregg Vanderheiden, Wendy Chisholm, John Slatin, and Jason White. 2008. Web content accessibility guidelines (WCAG) 2.0. *WWW Consortium (W3C)* 290, 1-34 (2008), 5–12.

[9] The Diagram Center. 2020. *Specific Guidelines – Graphs.* https://diagramcenter.org/specific-guidelines-e.html

[10] Jinyue Chen, Lingyu Kong, Haoran Wei, Chenglong Liu, Zheng Ge, Liang Zhao, Jianjian Sun, Chunrui Han, and Xiangyu Zhang. 2024. Onechart: Purify the chart structural extraction via one auxiliary token. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 147–155.

[11] Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588* (2022).

[12] Yufan Chen, Ruiping Liu, Junwei Zheng, Di Wen, Kunyu Peng, Jiaming Zhang, and Rainer Stiefelhagen. 2025. Graph-based Document Structure Analysis. *arXiv preprint arXiv:2502.02501* (2025).

[13] Zhi-Qi Cheng, Qi Dai, and Alexander G Hauptmann. 2023. Chartreader: A unified framework for chart derendering and comprehension without heuristic rules. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 22202–22213.

[14] Christopher Clark and Santosh Divvala. 2016. Pdffigures 2.0: Mining figures from research papers. In *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries*. 143–152.

[15] Android Developers. 2012. *Making Applications Accessible: contentDescription.* https://stuff.mit.edu/afs/sipb/project/android/docs/guide/topics/ui/accessibility/apps.html

[16] Android Developers. 2025. *Create your own accessibility service.* https://developer.android.com/guide/topics/ui/accessibility/service

[17] Madalina Dicua and Camelia Chiraa. 2025. Evaluating Deep Learning Models for Cross-Platform UI Component Detection: A Study Across Web, Desktop, and Mobile Interfaces. *Procedia Computer Science* 270 (2025), 2977–2986.

[18] Iyad Abu Doush, Enrico Pontelli, Dominic Simon, Tran Cao Son, and Ou Ma. 2009. Making Microsoft Excel™ accessible: Multimodal presentation of charts. ASSETS'09-Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility (2009), 147–154.

[19] Diogo Duarte, Rita Costa, Pedro Bizarro, and Carlos Duarte. 2024. AutoVizuA11y: A tool to automate screen reader accessibility in charts. In *Computer Graphics Forum*, Vol. 43. Wiley Online Library, e15099.

[20] Emory J Edwards, Michael Gilbert, Emily Blank, and Stacy M Branham. 2023. How the alt text gets made: What roles and processes of alt text creation can teach us about inclusive imagery. *ACM Transactions on Accessible Computing* 16, 2 (2023), 1–28.

[21] Ali Mazraeh Farahani, Peyman Adibi, Mohammad Saeed Ehsani, Hans-Peter Hutter, and Alireza Darvishy. 2023. Automatic chart understanding: a review. *IEEE Access* 11 (2023), 76202–76221.

[22] Google. 2025. *Develop UI for Android.* https://support.google.com/accessibility/android/answer/6283677

[23] Google Cloud. 2025. *Overview of Prompting Strategies.* https://docs.cloud.google.com/vertex-ai/generative-ai/docs/learn/prompts/prompt-design-strategies

[24] Richard W Hamming. 1950. Error detecting and error correcting codes. *The Bell system technical journal* 29, 2 (1950), 147–160.

[25] Kung-Hsiang Huang, Hou Pong Chan, May Fung, Haoyi Qiu, Mingyang Zhou, Shafiq Joty, Shih-Fu Chang, and Heng Ji. 2024. From pixels to insights: A survey on automatic chart understanding in the era of large foundation models. *IEEE Transactions on Knowledge and Data Engineering* 37, 5 (2024), 2550–2568.

[26] John D Hunter. 2007. Matplotlib: A 2D graphics environment. *Computing in science & engineering* 9, 03 (2007), 90–95.

[27] Philipp Jahoda. 2014. *MPAndroidChart: A Powerful Android Chart View / Graph View Library.* https://github.com/PhilJay/MPAndroidChart

[28] Ishan Amarkumar Joshi. 2021. Unblind the charts: Towards making interactive charts accessible in android applications. *arXiv preprint arXiv:2109.12442* (2021).

[29] Kaggle. 2025. *Datasets.* https://www.kaggle.com/datasets

[30] Gang Li and Yang Li. 2023. Spotlight: Mobile UI understanding using vision-language models with a focus.(2023).

[31] Fangyu Liu, Julian Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhu Chen, Nigel Collier, and Yasemin Altun. 2023. DePlot: One-shot visual language reasoning by plot-to-table translation. In *Findings of the Association for Computational Linguistics: ACL 2023*. 10381–10399.

[32] Fangyu Liu, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Yasemin Altun, Nigel Collier, and Julian Eisenschlos. 2023. Matcha: Enhancing visual language pretraining with math reasoning and chart derendering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 12756–12770.

[33] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. Ssd: Single shot multibox detector. In *European conference on computer vision*. Springer, 21–37.

[34] Alan Lundgard and Arvind Satyanarayan. 2021. Accessible visualization via natural language descriptions: A four-level model of semantic content. *IEEE transactions on visualization and computer graphics* 28, 1 (2021), 1073–1083.

[35] Junyu Luo, Zekun Li, Jinpeng Wang, and Chin-Yew Lin. 2021. Chartocr: Data extraction from charts images via a deep hybrid framework. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 1917–1925.

[36] Yuyu Luo, Xuedi Qin, Nan Tang, and Guoliang Li. 2018. Deepeye: Towards automatic data visualization. In *2018 IEEE 34th international conference on data engineering (ICDE)*. IEEE, 101–112.

[37] Yuyu Luo, Xuedi Qin, Nan Tang, Guoliang Li, and Xinran Wang. 2018. Deepeye: Creating good data visualizations by keyword search. In *Proceedings of the 2018 International Conference on Management of Data*. 1733–1736.

[38] Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*. 2263–2279.

[39] Ahmed Masry, Parsa Kavehzadeh, Xuan Long Do, Enamul Hoque, and Shafiq Joty. 2023. Unichart: A universal vision-language pretrained model for chart comprehension and reasoning. *arXiv preprint arXiv:2305.14761* (2023).

[40] Nitesh Methani, Pritha Ganguly, Mitesh M Khapra, and Pratyush Kumar. 2020. Plotqa: Reasoning over scientific plots. In *Proceedings of the ieee/cvf winter conference on applications of computer vision*. 1527–1536.

[41] Omar Moured, Morris Baumgarten-Egemole, Karin Müller, Alina Roitberg, Thorsten Schwarz, and Rainer Stiefelhagen. 2024. Chart4blind: An intelligent interface for chart accessibility conversion. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*. 504–514.

[42] Hui Ni and The ncnn contributors. 2025. *ncnn.* https://github.com/Tencent/ncnn

[43] Jason Obeid and Enamul Hoque. 2020. Chart-to-text: Generating natural language descriptions for charts by adapting the transformer model. *arXiv preprint arXiv:2010.09142* (2020).

[44] Pivotal Software, Inc. 2025. *Spring Boot.* https://spring.io/projects/spring-boot

[45] Yash Prakash, Pathan Aseef Khan, Akshay Kolgar Nayak, Sampath Jayarathna, Hae-Na Lee, and Vikas Ashok. 2024. Towards Enhancing Low Vision Usability of Data Charts on Smartphones. *IEEE Transactions on Visualization and Computer Graphics* (2024).

[46] Appium Project. 2025. *Welcome - Appium Documentation.* https://appium.io

[47] PyTorch. 2025. *PyTorch: an open source machine learning framework.* https://pytorch.org/

[48] Sebastián Ramírez and FastAPI Contributors. 2025. *FastAPI.* https://github.com/fastapi/fastapi

[49] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* 28 (2015).

[50] Ather Sharif, Andrew Zhang, and Jacob Wobbrock. 2025. Increasing agency of screen-reader users in consuming information from online data visualizations. In *Proceedings of the 22nd International Web for All Conference*. 147–159.

[51] Benny Tang, Angie Boggust, and Arvind Satyanarayan. 2023. Vistext: A benchmark for semantically rich chart captioning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 7268–7298.

[52] John R Thompson, Jesse J Martinez, Alper Sarikaya, Edward Cutrell, and Bongshin Lee. 2023. Chart reader: Accessible visualization experiences designed with screen reader users. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–18.

[53] Ultralytics. 2023. *YOLOv8: Ultralytics Official Documentation.* https://docs.ultralytics.com/models/yolov8/

[54] Ultralytics. 2024. *YOLOv11: Ultralytics Official Documentation.* https://docs.ultralytics.com/models/yolo11/

[55] Ultralytics. 2025. *YOLO12: Ultralytics Official Documentation.* https://docs.ultralytics.com/models/yolo12/

[56] Antony Unwin. 2020. Why is data visualization important? What is important in data visualization. *Harvard Data Science Review* 2, 1 (2020), 1.

[57] Ramarathnam Venkatesan, S-M Koon, Mariusz H Jakubowski, and Pierre Moulin. 2000. Robust image hashing. In *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)*, Vol. 3. IEEE, 664–666.

[58] W3C Web Accessibility Initiative (WAI). 2025. *Complex Images*. https://www.w3.org/WAI/tutorials/images/complex/

[59] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[60] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388* (2025).

[61] Qi Yang, Weichen Bi, Haiyang Shen, Yaoqi Guo, and Yun Ma. 2025. PixelWeb: The First Web GUI Dataset with Pixel-Wise Labels. *arXiv preprint arXiv:2504.16419* (2025).

[62] Zhuohao Zhang, John R Thompson, Aditi Shah, Manish Agrawal, Alper Sarikaya, Jacob O Wobbrock, Edward Cutrell, and Bongshin Lee. 2024. ChartA11y: Designing accessible touch experiences of visualizations with blind smartphone users. In *Proceedings of the 26th International ACM SIGACCESS Conference on Computers and Accessibility*. 1–15.