**ABSTRACT**

Chart images visually represent quantitative information. Most of these visual information are represented by graphical symbols and textual descriptions; without access to the Object Model of a graphic it is difficult for viewers to acquire the accurate underlying data. In response we propose the **VIEW** (Visual Information Extraction Widget), a system that automatically extracts information from raster-format charts to improve accessibility. Taking a chart image as input, the system first segments the image into connected-components and distinguishes them as graphical and textual components. By analyzing the graphical components, the system then identifies the graphic type and further conducts category-specific methods to infer the underlying data. Using the images drawn from the web, we conduct experiments to demonstrate the effectiveness of the proposed system. Based on the extracted information, **VIEW** generates a general-purpose descriptive data table, leading the production of multi-modal representations under the task-oriented design principle.

**Index Terms**— Chart image understanding, image processing, text detection, machine learning, accessibility.

**1. INTRODUCTION**

With the growth of electronic documents and the development of high-bandwidth digital communication systems, chart images (line graphs, bar charts, pie charts, etc.) are increasingly pervasive in digital media. Such graphics present a wide variety of quantitative information [1]. However, since most of these visual information are represented by graphical symbols and the correspondent textual descriptions in raster-format images, the accessibility to the underlying Object Model of a graphic is limited, which makes it difficult for viewers to specify the accurate numerical data and further reuse them across analysis tasks [2]. In order to improve information accessibility to the chart images, it is necessary to develop tools that automatically extract the underlying information from chart images and represent them in alternative accessible modalities [3].

Studies on the accessibility of the chart images have made significant progress recently. Motivated by the assumption that different image classes require different treatment in terms of information extraction, researchers have developed classification algorithms to identify the chart type [4]. Specifically, Prasad et al. [5] utilize high-level image features for classification, while Savva et al. [6] sample the image patches to learn low-level features for chart type identification. When it comes to the information extraction, both graphical and textual contents are analyzed: text detection techniques has been investigated in [2, 4], on the other hand, graphical information extraction is considered (e.g., edge-map based methods [2, 3, 7] and connect-component based method [6]).

Once the contained information is extracted, alternate representations are generated according to the user preference. Chart redesign has been investigated for effective visualization [6, 8, 9]. Visual-Assitive techniques, which enable visual impaired users to explore the graphical information, provide
alternate modalities such as tactile graphics, haptic devices and interactive audio systems [4, 10].

In this paper, we propose VIEW (Visual Information Extraction Widget), a systematic approach that automatically extracts comprehensive information contained in a chart image. The system takes raster-format chart images as input and generates a general-purpose data table. Therefore, the system can lead the production of multi-modal representations under the task-oriented design principle, for example, with the extracted data, we can convert the static chart image (Figure 1 (left)) to an interactive graphic (Figure 1 (right)), which provides accurate numerical data for users. Noted that the extracted data can be also reused across analysis tasks and platforms.

2. SYSTEM OVERVIEW

The system VIEW proposed in this paper is aimed to extract the underlying information from chart images. At this moment, we limit the scope of our studies to three classes of non-continuous-textured 2D chart images, specifically, bar charts, pie charts and line graphs. The proposed systematic approach includes three stages: first, for each input raster graphic, the system utilizes intensity and morphological attributes to segment the image into a set of connected components, then discriminates textual and graphical components via image processing and machine learning techniques; second, it extracts geometric features from the graphical components, and applies a Support Vector Machine (SVM) for chart classification; and finally, the system conducts category-specific information extraction to infer the underlying data. Once the system extracts information, a general-purpose data table is generated for accessible applications.

2.1. Image Segmentation and Text Detection

Given an information graphic, the system incorporates intensity and morphology attributes to segment it into a set of connected components, which can be further discriminated into the textual and graphical components.

As noted above, the texture of charts considered in this study are non-continuously toned, the graphical symbols and the texts are solidly shaded using a single color, which suggests that the distribution of the histogram of pixel intensities is highly sparse. In order to detect the comprised intensity layers, we extend the robust non-parametric clustering algorithm developed in [11]. Suppose the histogram of a graphic with 8-bit grayscale intensity depth is given by \( G = \{ g(i) | i \in [0, 255] \} \), where \( g(i) \) is the number of pixels with intensity level \( i \) (without losing generality, RGB images consists of 3-dimension vectors.). The distance between any two intensity levels is defined as \( \delta(i, j) = \exp(-|i - j|) \). Then, the prototypes are classiﬁed as graphical components and they are analyzed in the following modules.

2.2. Image Classification

In this section, we extract geometric features from the graphical components (summarized in Table 1), to classify the chart images. A graphical connected component consists of solid-filled and/or thin-line segments. Typically, the solid-filled segments are used to represent the symbols and to list legend entries; while thin-line segments are invoked to draw the intended data in line graphs as well as to display axes, reference lines, etc., in all chart images. Noted that, in a bar/pie chart, the segments representing the intended data can be either solid-filled or thin-line segments, which are both considered in this work. Specifically, we apply the erosion and dilation operations [12] to the components, decomposing the components into solid-filled and thin-line segments.

For a solid-filled segment, the shape of the segment can be identified based on the structure. Noting that if the area, measured in pixels, of a rectangular segment is smaller than a threshold, the rectangular segment is treated as a legend entry and not utilized in the graphic category identification. For all the non-legend solid-filled segments, a shape list, descendingly sorted according to the area of the segments (i.e., the

<table>
<thead>
<tr>
<th>Graphic Type</th>
<th>Symbol</th>
<th>Line Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bar chart</td>
<td>rectangles</td>
<td>straight line</td>
</tr>
<tr>
<td>pie chart</td>
<td>sectors</td>
<td>straight line; arc</td>
</tr>
<tr>
<td>line graph</td>
<td>lines</td>
<td>straight line</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 1. Basic components in the chart images</th>
</tr>
</thead>
</table>

For all the non-legend solid-filled segments, a shape list, descendingly sorted according to the area of the segments (i.e., the
number of pixels involved in a solid-filled segment), is formed as \( \{ S_i \}_{i=1}^{N_s} \), where \( N_s \) is the number of solid-filled segments in an information graphic and \( S_i \) holds 1 for a rectangle or \(-1\) for a sector.

Concurrently, for all \( N_l \) thin-line segments, we formulate each instance, \( l_i \), into a list of overlapped \( 3 \times 3 \) patches \( \{ p_{ij} \}_{j=1}^{N_l} \), where \( n_i \) is the number of pixels of \( l_i \), and the patches are centered on each line pixel. It is known that different type of graphics consist of different frequently occurring line patterns [6], such as vertical and horizontal lines in bar charts, arcs in pie charts, and lines with various slopes in line graphs. For a \( 3 \times 3 \) patch, there are at most \( 2^8 \) patterns. Therefore, We then sum the activations [6] of each patch pattern \( \{ \{ a p_i \} [i \in [1, 2^8]] \} \) of all the \( N_l \) thin line segments.

Given the lists of \( \{ S_i \}_{i=1}^{N_s} \), the first feature, \( t_{k0} \), is defined as

\[
t_{k0} = \left\{ \frac{\sum_{i=1}^{N_s} 0.5^i S_i}{t_{k1}}, \quad \text{for } N_s \geq 1 \right\} 
\]

\[
= 0, \quad \text{for } N_s = 0
\]

where \( 0.5^i \) is the weight for \( S_i \). The line features comprise of 256 elements, and each one is normalized as \( t_{ki} = \frac{\sum_{i=1}^{N_l} a p_i}{\sum_{i=1}^{2^8} a p_i} \), where \( i \in [1, 256] \).

Finally, we use the 257-dimensional feature vector to perform classification using SVM [14].

### 2.3. Information Extraction and Data Depiction

By applying the previous two modules, the system obtains the textual components, the graphical components and the graphic type. At this stage, they are jointly analyzed for the information extraction. More specifically, the system addresses the graphic understanding issues by inferring both textual and graphical information. Based on the extracted information, the system enables a general-purpose data table to describe the chart image.

To infer the information of the text blocks, the system first applies the OCR software to recognize the semantic information. Also noted that text blocks indicate specific descriptive information according to their logic roles (captions, labels, etc.), the proposed system utilizes a Naive Bayes classifier for parsing the logic roles of the text blocks. Specifically, assuming that each text block \( x \) can be described by attributes \( < a_1, a_2, \ldots, a_n > \), the most probable logic tag \( v \) mapped by \( f(x) \) is

\[
v_{MAP} = \arg \max_{v_j \in V} P(a_1, a_2, \ldots, a_n | v_j) P(v_j),
\]

where \( P(v_j) \) is the prior about how often a specific tag shows up in a graphic, and the \( P(a_1, a_2, \ldots, a_n | v_j) \) is the likelihood function that we can learn from the training data. At this moment, simplified attributes such as the block size, the block position are used in the likelihood function.

For the graphical objects, it is necessary to identify symbols which convey the underlying data and further infer the information from the geometric patterns of the symbols. It is noted that the graphics of the type share the common elements, and thus this procedure are highly category-specific. In this paper, some general category constraints for parsing the graphical information are listed for notification, the detailed rules can be found in [13].

**Pie charts** employ sectors to illustrate proportion, and thus the proportion of each arc can be calculated for the proportion. For **bar charts**, it is important to first identify the orientations of the bar symbols, and then the height of each bar symbol is identified for data inferring. In **line graphs**, the intended data is represented by one or more curves. In order to identify the line symbol, it is important to discriminate the line symbols from the axes or grid lines. Once the line symbols are extracted, the relative position of the line point can be further inferred.

After acquiring graphical symbols and the text blocks, the system associates the numerical data from the symbols with the corresponding text blocks to infer the underlying data according to the category-specific constraint grammar. And finally, it is necessary to translate the information graphic from its original image format to a data table, which describes the graphic in terms of its essential properties, for the other representation modules. Table 2 shows a preliminary general-purpose descriptive data table. The description framework is dissected into two levels. The terms **cht** and **chs** describe the overall information regarding to the graphic, while **che** and **chd** detail the underlying information in the graphic. Noting that we do not include the text information in the data table, but the evaluation of the logic role identification will be presented in 3.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cht</td>
<td>graphic type.</td>
</tr>
<tr>
<td>chs</td>
<td>graphic size. ([\text{height}, \text{width}]^2).</td>
</tr>
<tr>
<td>chv</td>
<td>variable number.</td>
</tr>
<tr>
<td>chd</td>
<td>data strings. One string for each variable.</td>
</tr>
</tbody>
</table>

### 3. SYSTEM EVALUATION AND APPLICATIONS

In this section, experiments are conducted to show the performance of the proposed system. The chart images employed in the experiments consist of 100 bar charts, 100 pie charts and 100 line graphs collected from various real-world digital resources. The image segmentation and text detection are detailed evaluated in [13]. In this paper, We first evaluate the performance of the image classification. Given the extracted features, the SVM classifier [14] identifies the chart type. By using the multi-class SVM in a 5-fold experiment, the classifier accuracy achieves 97.3\%, especially, all the pie chart are correctly identified.

We further evaluate the logic role identification for text
blocks. Particularly, we test the algorithm by detecting the graphic caption. To train the Bayesian model, we select 25 graphics. By counting the appearance of the caption blocks, the prior probability, \( P(v_j) \), is calculated. During the testing phase, 20 graphics are randomly chosen, which contain more than 300 text blocks. VIEW identified all the caption, while only misclassified 3 non-caption blocks as captions.

With the extracted information, VIEW enables a general-purpose graphic descriptive data table (illustrated in Fig 2). This scheme can further lead to multiple representation modalities according to the task-oriented principle. As shown in Fig. 1, a static chart image is converted to an interactive graphic. This application benefits the computer users, where they can explore the chart images by pointing at specific region to acquire the numerical data with ease. This explore manner can even facilitate the highly growing mobile device users, noting that the limited screen size may hinder the users to acquire the detailed graphic in high-resolution, by leveraging the multi-touch screen and the audio system, the interactive graphics provide user with data more efficiently.

The extracted data are also reusable across data analysis tasks and platforms. Considering the circumstance that researchers collected 5-years annual market share (5 pie charts similar to Fig. 1 (Left)), and they would like to compare the share trend for a specific product, it is then convenient to draw the extracted data by VIEW and plot a trend graph.

### 4. CONCLUSION AND FUTURE WORK

In this paper we propose VIEW, a system for robustly and effectively extracting the underlying data from bar charts, pie charts and line graphs. Operating on information graphics in raster format, the proposed methodology achieves 97\% accuracy in identifying the graphic types, and robustly extract both textual and graphical information. Based on the extracted information, we also demonstrate the data table and further develop practical applications.

However, more work is needed to improve the performance of system. First, in order to advance the image classification, additional features need to be developed (such as taking the text blocks features into account). Moreover, in the text-format data table, more parameters need to be generated to fully describe the graphics, for example, the additional textual information and the axes. Finally, the capacity of the system can be improved by extending the image class and considering 3D graphics.

### 5. REFERENCES


