dation set contained in ExcelChart400K is used to set the hyper-parameters. We use the early-stopping strategy for the model training.

6. Evaluation Metric

In previous works, researchers usually borrow evaluation metrics from other domains, e.g., object detection or information retrieval. Those methods do not take into account the specialty of chart data. In this paper, we propose three evaluation metrics for three chart types.

6.1. Bar Chart

For bar chart inputs, our goal is to match the bounding box $p = [x_p, y_p, w_p, h_p]$ to the ground truth bounding box $g = [x_g, y_g, w_g, h_g]$. First, we define a custom distance function for pairwise-wise point comparisons:

$$D(p, g) = \min(1, \| \frac{x_p - x_g}{w_g} \| + \| \frac{y_p - y_g}{h_g} \| + \| \frac{h_p - h_g}{h_g} \|) \tag{9}$$

Here we only consider the differences between x, y, h because w is not related to chart reading. Then we compute the pairwise cost matrix $\mathbf{C}$, where $\mathbf{C}_{n,m} = D(p_n, g_m)$. Then we can find the minimum total cost by taking it as the job-assignment problem:

$$cost = \min_{\mathbf{X}} \sum_i^K \sum_j^K \mathbf{C}_{i,j} \mathbf{X}_{i,j} \tag{10}$$

Then the score can be defined as $score = 1 - cost/K$, where $K = \max(N, M)$. $\mathbf{X} \in \{0, 1\}$ is a binary assignment matrix since each point will only be assigned once.

6.2. Line Chart

Since a line defines a sequence of continuous data, we treat it as a continuous similarity problem. Let $P = [(x_1, y_1), ..., (x_N, y_N)]$ be the predicted point set and $G = [(u_1, v_1), ..., (u_M, v_M)]$ be the ground-truth set. We define the average error rate between the ground-truth point set G to the predicted point set P using precision and recall

$$Prec(P, G) = Rec(G, P) \tag{11}$$

$$Rec(P, G) = \frac{\sum_{i=1}^M (1 - Err(v_i, u_i, P)) * Intv(i, G)}{u_M - u_1} \tag{12}$$

$$F1 = 2 \cdot Prec \cdot Rec(Prec + Rec) \tag{13}$$

where $Err(v_i, u_i, P)$ defines error rate of matching point (u_i, v_i) for point set P. $Intv(i, G)$ defines the ratio of the ith point in final score. More specifically,

$$Err(v_i, u_i, P) = \min(1, \| \frac{v_i - I(P, u_i)}{v_i} \|) \tag{14}$$

$$Intv(i, G) = \begin{cases} \frac{u_{i+1} - u_i}{2} & \text{for } i = 1 \\ \frac{u_i - u_{i-1}}{2} & \text{for } i = M \\ \frac{u_{i+1} - u_{i-1}}{2} & \text{for } 1 < i < M \end{cases} \tag{15}$$

Here $I(P, u_i)$ is a linear interpolation function that computes the value of line P at the point u_i. The $Err(v_i, u_i, P)$ ranges from 0 to 1. The $Intv(i, G)$ ratio will arise if the gap gets bigger between two points as described in Eq. (15). If there are multiple lines in one chart, we will enumerate the combinations to find the best match score.

6.3. Pie Chart

Both the data values and the ordering are important for pie chart reading. For Pie Chart images we consider the data extraction as a sequence matching problem. Let $P = [x_1, ..., x_N]$ be the predicted data sequence in clockwise order and $G = [y_1, ..., y_M]$ be the ground-truth data sequence. Then the matching $score(N, M)$ can be defined as:

$$\begin{aligned} score(i, j) = max(&score(i-1, j), score(i-1, j), \\ &score(i-1, j-1) + 1 - \| \frac{x_i - y_j}{y_j} \|) \end{aligned} \tag{16}$$

$$score = \frac{score(N, M)}{M} \tag{17}$$

Where $\forall i \ score(i, 0) = 0$, $\forall j \ score(0, j) = 0$. The score is obtained through dynamic programming.

7. Experiment

7.1. Baseline Methods

We compare our method with three types of methods: rule-based methods, deep-learning-based methods and off-the-shelf commercial product. For rule-based methods, **Revision**[23] is a model capable for data extraction of bar chart and pie chart. For deep-learning-based methods, we report the performance of **Vis** [6] on the public datasets and ExcelChart400K dataset. We also implement **ResNet+Faster-RCNN** which is an enhanced version of [6, 17] with Faster-RCNN [22] using ResNet [9] backbone for bar chart extraction, and **Rotation RNN** [17] for pie chart extraction. In addition, we report performance on **ResNet+RNN**, a fully end-to-end deep RNN approach as a strong baseline. In this model, after the deep feature extraction network, RNN is directly applied to output the desired data in the sorted order. For commercial product, we use **Think Cell**[2] which is capable for bar chart data extraction. We can only access it

[2]https://www.think-cell.com/