

# Supplementary Information

## 1. More Details about Cycle Centernet Arcitecture

As shown in Fig. 1, a wired table image will be sent into backbone to get a feature map  $F \in R^{\frac{h}{4} \times \frac{w}{4} \times c}$ , then we use Keypoint detection module and Cycle-Pairing module to detect the cells and learn splicing information. Keypoint detection module is a common detection module following Centernet, it will get a heatmap  $Y \in R^{\frac{h}{4} \times \frac{w}{4} \times 2}$  and an offset  $O \in R^{\frac{h}{4} \times \frac{w}{4} \times 2}$  to locate each cell center point and vertex. It uses the same loss function: keypoint regression and offset loss  $L_{det}$  to monitor its accuracy, more detail can be found in Centernet. And the Cycle-Pairing module with Pairing loss will get the splicing information  $VC_{map}$  and  $CV_{map}$  of all the cells. From the output of Cycle Centernet, we can get a complex table structure which has cells' coordinates information. Finally, we use Parsing-Processing module to get the row and column number of each cell.

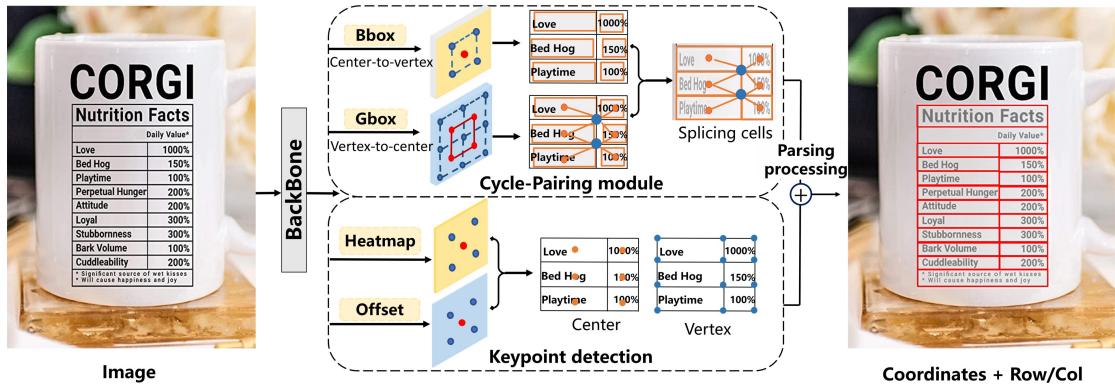


Figure 1. Network Arcitecture

## 2. Parsing-Processing Module

The Parsing-Processing module is the last step to parsing cell row and column information from a complete table. Below is the pseudo codes mentioned in 3.2 and 4.3.

---

**Algorithm 1: Calculating logical coordinates of cells.**

---

```
1 Input: a set of table cells  $C = \{c^i\}_{i=1}^{|C|}$ , where  $c^i = (pt^{ik}), k \in [1,4]$ .
2 Output: a set of logical coordinates of cells  $LC = \{lc^i\}_{i=1}^{|C|}$ .
3 for each  $c^i \in C$  do
4    $leftedge^i = [pt^{i0}, pt^{i3}], rightedge^i = [pt^{i1}, pt^{i2}]$ .
5    $upedge^i = [pt^{i0}, pt^{i1}], downedge^i = [pt^{i3}, pt^{i2}]$ .
6 end for
7 # merging edges sharing same  $pt$  into lines
 $E_v = MERGE(\{leftedge_i\}_{i=1}^{|C|}, \{rightedge_i\}_{i=1}^{|C|}), E_h = MERGE(\{upedge_i\}_{i=1}^{|C|}, \{downedge_i\}_{i=1}^{|C|})$ .
8 sort  $E_v$  according to x coordinates.
9 sort  $E_v$  according to y coordinates.
10 for each  $c^i \in C$  do
11    $startrow^i = INDEX(upedge^i \rightarrow E_h)$ .
12    $endrow^i = INDEX(downedge^i \rightarrow E_h) - 1$ .
13    $startcolumn^i = INDEX(leftedge^i \rightarrow E_v)$ .
14    $endcolumn^i = INDEX(rightedge^i \rightarrow E_v) - 1$ .
15    $lc^i = (startrow^i, endrow^i, startcolumn^i, endcolumn^i)$ 
16 end for
```

---

First, it split every cell into 4 bounding edges, then merge the up edges and down edges to horizontal lines and merge left edges and right edges to vertical lines according to cell connectivity. Next, it sort horizontal lines, vertical lines and index them from 0. Finally, it ranks cells by line index and outputs row/column information. The detailed pseudo code will be given in the supplementary materials.

## 3. The heuristic grouping scheme

The heuristic grouping scheme is mentioned in 3.2 Baselines and Benchmark Evaluation, which is designed to splice all the discrete cells into structured digital tables. Below is the pseudo codes.

---

**Algorithm 2: Grouping discrete cells into structured digital tables.**  $C = \{c^i\}_{i=1}^{|C|}$  is a set of table cells that composed of width,height and four vertex coordinates, where  $c^i = (w^i, h^i, v^{ik}), v^{ik} = (x^{ik}, y^{ik}), k \in [1,4]$ .

---

```

1  Input:  $C = \{c^i\}_{i=1}^{|C|}$ .  $r = 0.1$ ,  $dis_{thresh} = 30$ .
2  Output:  $C = \{c^i\}_{i=1}^{|C|}$ . Vertex coordinates in C are updated.
3  for each  $c^i \in C$  do
4    for each  $v^{ik} \in c^i$  do
5       $xoffset1 = \max(w^i * r, 4.0)$ ,  $yoffset1 = \max(h^i * r, 4.0)$   $keep = []$ 
6      for each  $c^j \in C$  but  $j \neq i$  do
7         $xoffset2 = \max(w^j * r, 4.0)$ ,  $yoffset2 = \max(h^j * r, 4.0)$ 
8        for each  $v^{jl} \in c^j$  do
9           $xdist = |x^{jl} - x^{ik}|$ ,  $ydist = |y^{jl} - y^{ik}|$ ,  $dist = \sqrt{(x^{jl} - x^{ik})^2 + (y^{jl} - y^{ik})^2}$ 
10         if  $xdist > xoffset1$  or  $xdist > xoffset2$  or  $ydist > yoffset1$  or  $ydist > yoffset2$  or  $dis > dis_{thresh}$  then
11           continue
12         else
13           Push the index of  $v^{jl}$  in  $keep$ 
14       end for
15     end for
16     Push the index of  $v^{ik}$  in  $keep$ 
17     Calculate the mean value of all vertex coordinates recorded in  $keep$ , and assign the mean value
as the vertex coordinates of corresponding cells in C.
18   end for
19 end for
20 The vertex not refined in C are spliced to the nearest vertex.

```

---

We need to splice the discrete cells twice, step 1: for each vertex of cells, calculate the distance from all the vertices of other cells to this vertex, if the distance is within a certain range, calculate the mean value of these vertex coordinates, and then assign the value to all these vertices. Step 2: for all the points that have not been refined in step 1, find the nearest refined vertex, if a vertex satisfies the condition in step 1, the points that have not been refined are spelled onto the output from step 1.