Symbol as Points: Panoptic Symbol Spotting via Point-based Representation

Wenlong Liu[1] Tianyu Yang[1] Yuhan Wang[2] Qizhi Yu[2] Lei Zhang[1]
[1]International Digital Economy Academy (IDEA) [2]Vanyi Tech

Code & Data & Model: https://github.com/nicehuster/SymPoint

Problem Definition and Contribution

Goal: Spotting and parsing both countable things(windows, doors, tables, etc.) and uncountable stuff(wall, railing, etc.) from CAD drawings.

Contribution:

- We carefully analyzing the data characteristics of CAD drawings and design novel and effective way of transferring CAD graphical primitives into 2d point sets and utilize methodologies from point cloud analysis to tackle the task of panoptic symbol spotting.
- We propose Attention with Connection Module (ACM), Contrastive Connection Learning (CCL) module and KNN Interpolation to effectively promote the performance of the model.
- Our approach outperforms recent SOTA method GAT-CADNet by a large margin on the FloorPlanCAD dataset.

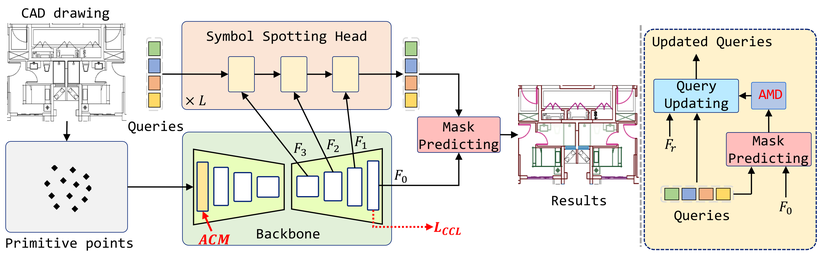
Problem Formulation

Task Model: Given a CAD drawing represented by a set of graphical primitives $\{p_k\}$, the *panoptic symbol spotting* task requires a map $F_p : p_k \mapsto (l_k, z_k) \in \mathcal{L} \times \mathbf{N}$, where $\mathcal{L} := \{0, \ldots, L-1\}$ is a set of predetermined set of object classes, and $\mathbf{N}$ is the number of possible instances. The semantic label set $\mathcal{L}$ can be partitioned into stuff and things subsets, namely $\mathcal{L} = \mathcal{L}^{st} \cup \mathcal{L}^{th}$ and $\mathcal{L}^{st} \cap \mathcal{L}^{th} = \emptyset$.

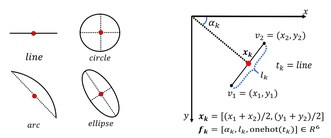
Main Idea: We first consider each graphic primitive as 2d point sets, and then utilize methodologies from point cloud analysis for graphic primitive representation learning.

Method

Network Architecture:

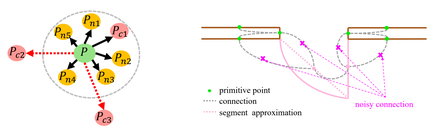


From symbol to points:



- Primitive position (left)
- Primitive feature (right)

ACM and Noisy Connection:



- Attend to connections (left)
- Noisy connections (right)

CCL and KNN Interpolation:

$$L_{CCL} = -\log \frac{\sum_{j \in \mathcal{A}(i) \wedge l_j = l_i} \exp(-d_{(i,j)}/\tau)}{\sum_{k \in \mathcal{A}(i)} \exp(-d_{(i,k)}/\tau)}, \quad A^r{}_{(j)} = \frac{\sum_{j \in \mathcal{K}(i)} A^0{}_{(j)}/d_{(i,j)}}{\sum_{j \in \mathcal{K}(i)} 1/d_{(i,j)}}$$

- Use L_{CCL} to alleviate the impact of noise connections(left)
- Use KNN interpolation to downsample attention masks(right)

Experiments & Results

Quantitative Results in FloorplanCAD:

Methods	PanCADNet (Fan et al., 2021)	CADTransformer (Fan et al., 2022)	GAT-CADNet (Zheng et al., 2022)	PointT‡ (Zhao et al., 2021)	**SymPoint (ours)**
F1	80.6	82.2	85.0	83.2	**86.8**
wF1	79.8	80.1	82.3	80.7	**85.5**

Table : **Semantic Symbol Spotting** comparison results with previous works. ‡: backbone with double channels. wF1: length-weighted F1.

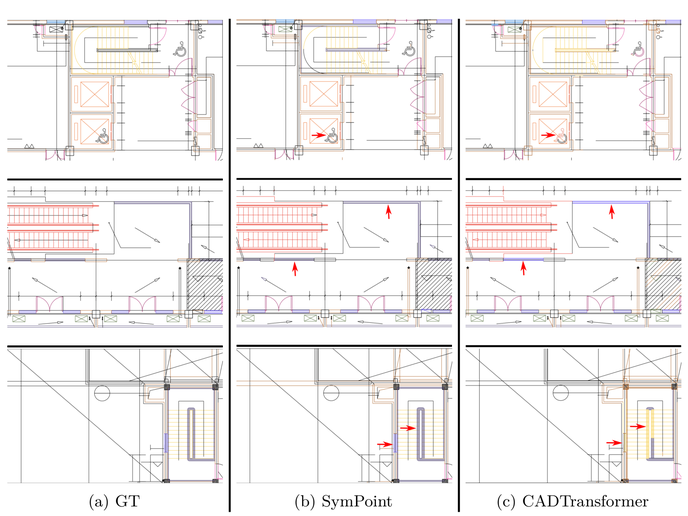
Method	Backbone	AP50	AP75	mAP	#Params	Speed
FasterRCNN (Ren et al., 2015)	R101	60.2	51.0	45.2	61M	59ms
YOLOv3 (Redmon & Farhadi, 2018)	DarkNet53	63.9	45.2	41.3	62M	11ms
FCOS (Tian et al., 2019)	R101	62.4	49.1	45.3	51M	57ms
DINO (Zhang et al., 2022)	R50	64.0	54.9	47.5	47M	42ms
SymPoint (ours)	PointT‡	**66.3**	**55.7**	**52.8**	35M	66ms

Table : **Instance Symbol Spotting** comparison results with image-based detection methods.

Method	Data Format	PQ	SQ	RQ	#Params	Speed
PanCADNet (Fan et al., 2021)	VG + RG	55.3	83.8	66.0	>42M	>1.2s
CADTransformer (Fan et al., 2022)	VG + RG	68.9	88.3	73.3	>65M	>1.2s
GAT-CADNet (Zheng et al., 2022)	VG	73.7	91.4	80.7	-	-
PointT‡Cluster(Zhao et al., 2021)	VG	49.8	85.6	58.2	31M	80ms
SymPoint(ours, 300epoch)	VG	**79.6**	89.4	**89.0**	35M	66ms
SymPoint(ours, 500epoch)	VG	**81.9**	90.6	**90.4**	35M	66ms
SymPoint(ours, 1000epoch)	VG	**83.3**	**91.4**	**91.1**	35M	66ms

Table : **Panoptic Symbol Spotting** comparisons results with previous works. VG: vector graphics, RG: raster graphics.

Qualitative Results in FloorplanCAD:



(a) GT (b) SymPoint (c) CADTransformer

Figure : Qualitative comparison of panoptic symbol spotting results with CADTransformer.

Quantitative Results in SESYD:

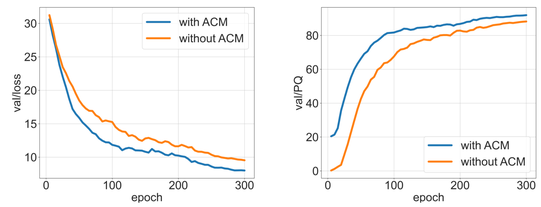
Methods	AP50	AP75	mAP
Yolov4	93.04	87.48	79.59
YOLaT	98.83	94.65	90.59
RendNet	98.70	98.25	91.37
SymPoint	96.79	95.63	91.01

(a) Performance comparison on floorplans.

Methods	AP50	AP75	mAP
Yolov4	88.71	84.65	76.28
YOLaT	96.63	94.89	89.67
RendNet	-	-	-
SymPoint	97.0	94.51	90.26

(b) Performance comparison on diagrams.

Table : **Performance comparison** on floorplans and diagrams.



(a) Loss curve. (b) PQ curve.

Figure : Convergence curves with/without the ACM Module on SESYD-floorplans.

Qualitative Results in SESYD:

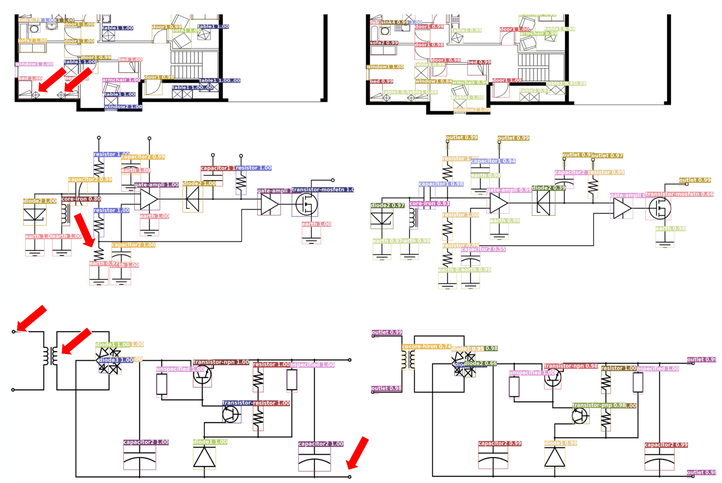
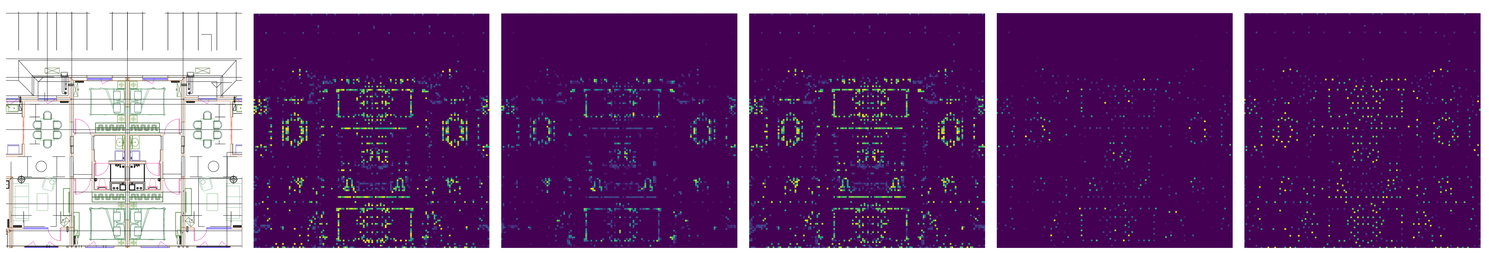


Figure : Qualitative comparison on floorplans and diagrams with YOLaT. The left column displays YOLaT's results, while the right column showcases ours.

Downsampling visualization for KNN interp vs bilinear interp:



Prediction | Score map | Bilinear interp (4x) | KNN interp (4x) | Bilinear interp (16x) | KNN interp (16x)