Class-Agnostic Visio-Temporal Scene Sketch Semantic Segmentation

Aleyna Kütük
KUIS AI Center
Koç University, İstanbul, Turkey
akutuk21@ku.edu.tr

Tevfik Metin Sezgin
KUIS AI Center
Koç University, İstanbul, Turkey
mtsezgin@ku.edu.tr

Abstract

Scene sketch semantic segmentation is a crucial task for various applications including sketch-to-image retrieval and scene understanding. Existing sketch segmentation methods treat sketches as bitmap images, leading to the loss of temporal order among strokes due to the shift from vector to image format. Moreover, these methods struggle to segment objects from categories absent in the training data. In this paper, we propose a Class-Agnostic Visio-Temporal Network (CAVT) for scene sketch semantic segmentation. CAVT employs a class-agnostic object detector to detect individual objects in a scene and groups the strokes of instances through its post-processing module. This is the first approach that performs segmentation at both the instance and stroke levels within scene sketches. Furthermore, there is a lack of free-hand scene sketch datasets with both instance and stroke-level class annotations. To fill this gap, we collected the largest Free-hand Instance- and Stroke-level Scene Sketch Dataset (FrISS) that contains 1K scene sketches and covers 403 object classes with dense annotations. Extensive experiments on FrISS and other datasets demonstrate the superior performance of our method over state-of-the-art scene sketch segmentation models. Our code and dataset can be accessed from https://github.com/aleynakutuk6/CAVT.

1. Introduction

Sketching is a rapid and widely adopted way for humans to visually express ideas. Especially with the rise of touch-screen technology, understanding hand-drawn sketches has become an essential task in the field of human-computer interaction. The field of sketch understanding includes various tasks such as sketch recognition, sketch-based image retrieval, and sketch segmentation. Sketch semantic segmentation stands out as a pivotal task, offering broad applicability in the analysis of sketches and facilitating tasks like sketch-based image retrieval. Despite the considerable attention given to semantic segmentation in natural images [1, 4, 18], this task remains relatively underexplored in sketches. Earlier studies on sketch segmentation have mostly concentrated on segmenting single-object sketches into semantically meaningful parts [14,16,24,28,33,34]. On the other hand, recent attention has shifted towards scene-level sketch semantic segmentation [2, 9, 21, 27, 32, 35].

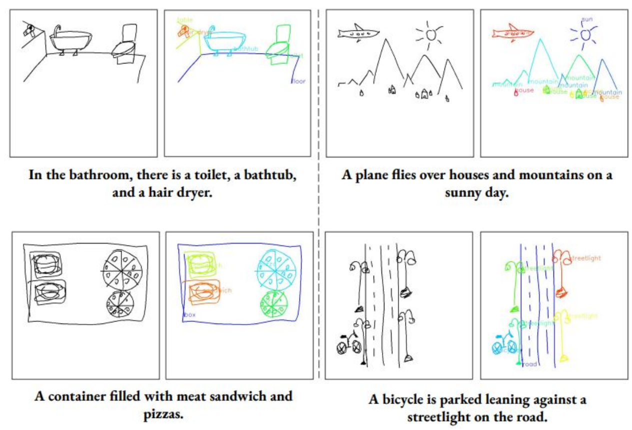

Figure 1. Sample scene sketches from FrISS dataset, each paired with corresponding textual scene descriptions. For each pair, the left image shows the black-and-white sketch, while the right image highlights the instance and stroke-level class annotations.

Sketches are processed either as stroke sequences or bitmap images. Many methodologies treat sketches as images and address sketch segmentation similarly to image segmentation tasks [2, 9, 21, 27, 35]. However, this direct approach often leads to the loss of temporal stroke information. As sketches consist of stroke sequences, capturing the stroke order can significantly enhance semantic segmentation performance. Moreover, current research on scene sketch segmentation mainly focuses on assigning a class to each pixel or stroke within a scene, thus segmenting scene sketches at the class level. Unfortunately, these methods cannot distinguish between individual objects that belong to the same class, such as two zebra instances in the same scene. To overcome these limitations, we introduce the

Dataset	# of Sketches	# of Cat.	Vector	Free-hand	Scene-level	Publicly Available	Annot. Type
QMUL Shoe [29]	419	1		✓		✓	C, I
QMUL Chair [29]	297	1		✓		✓	C, I
Sketchy [20]	75K	125	✓	✓		✓	C, I
TU-Berlin [7]	20K	250	✓	✓		✓	C, I
QuickDraw [12]	50M+	345	✓	✓		✓	C, I
SketchyScene [35]	7K+	45			✓	✓	C, I
SketchyCOCO [8]	14K+	17			✓	✓	C, I
SKY-Scene [9]	7K+	30			✓	✓	C
TUB-Scene [9]	7K+	35			✓	✓	C
CBSC [31]	331	74	✓	✓	✓	✓	C, I
FS-COCO [5]	10K	92-150	✓	✓	✓	✓	D
SFSD [32]	12K+	40	✓	✓	✓		C
FrISS (Ours)	1K	403	✓	✓	✓	✓*	C, D, I

Table 1. Summary of the sketch datasets. C, I, and D denote class-level annotations, instance-level annotations, and scene sketch textual descriptions, respectively. ✓*: the dataset will be publicly available after acceptance.

Class-Agnostic Visio-Temporal Network (CAVT) that processes scene sketches and generates stroke-level groupings of instances without relying on predefined class labels. Our approach leverages visual information via an object detector and incorporates the temporal order of strokes using both a post-processing module and an RGB coloring technique.

The primary challenge for scene sketch semantic segmentation lies in the absence of large-scale scene sketch datasets. Existing scene sketch datasets are typically constructed by inserting pre-defined clip-art or free-hand single-instance sketches into the layouts of reference images [8, 9, 35]. These datasets preserve the scene sketches in image format, limiting their utilization in stroke-based sketch methods. More recently, scene datasets have been collected by instructing participants to draw scenes based on reference natural images [5, 32]. However, this often results in the loss of participants' natural drawing behavior, as individuals tend to replicate the object positions and postures from the reference images.

In this work, we collected the largest Free-hand Instance- and Stroke-level Scene Sketch Dataset (FrISS), consisting of free-hand scene sketches in vector format, accompanied by textual descriptions, verbal audio recordings, and annotations at both the stroke and instance levels. To capture natural drawing behavior, participants were provided only with textual scene descriptions during the drawing process, without being shown any reference images. This approach ensures that FrISS features a diverse range of scene sketches that are not mere copies of reference images. Moreover, we avoided prolonged drawing sessions or multiple attempts, thus preventing artificially polished scene sketches. In summary, our main contributions are highlighted as follows:

1. We propose CAVT, a novel scene sketch semantic segmentation pipeline, that utilizes both visual and temporal information in the scene. This is the first study on scene sketch semantic segmentation that works at both instance and stroke levels.

2. We introduce FrISS, a densely annotated dataset that includes 1K free-hand scene sketches covering 403 object categories. FrISS can promote future stroke-based scene-level studies.

3. We conduct extensive experiments on FrISS and other free-hand scene sketch datasets and show that our approach achieves state-of-the-art performance.

2. Related Work

2.1. Sketch Semantic Segmentation

Existing works on sketch semantic segmentation mostly focus on single-object sketch datasets and divide an object into its semantically valid parts [14, 16, 24, 28, 33, 34]. On the other hand, scene-level sketch semantic segmentation aims to distinguish individual object instances within the scene. Regarding the processing of sketches, these studies can be divided into two main groups: image-based and sequence-based. Image-based methods typically treat sketches as raster images and output pixel-level segmentation predictions; whereas sequence-based methods utilize stroke-level information and assign semantic labels to each stroke in a sketch. Even if the majority of studies on single-object sketch semantic segmentation lie in the sequence-based methods [14, 24, 28, 33], there are not many studies conducted on stroke-level scene sketch semantic segmentation. This is mostly due to the lack of large-scale scene sketch datasets with stroke-level class annotations.

Prior works on scene sketch semantic segmentation treat the task as a semantic image segmentation problem, disregarding the stroke order [2, 9, 27, 35]. SketchyScene [35] is the pioneering study that assigns object categories at the

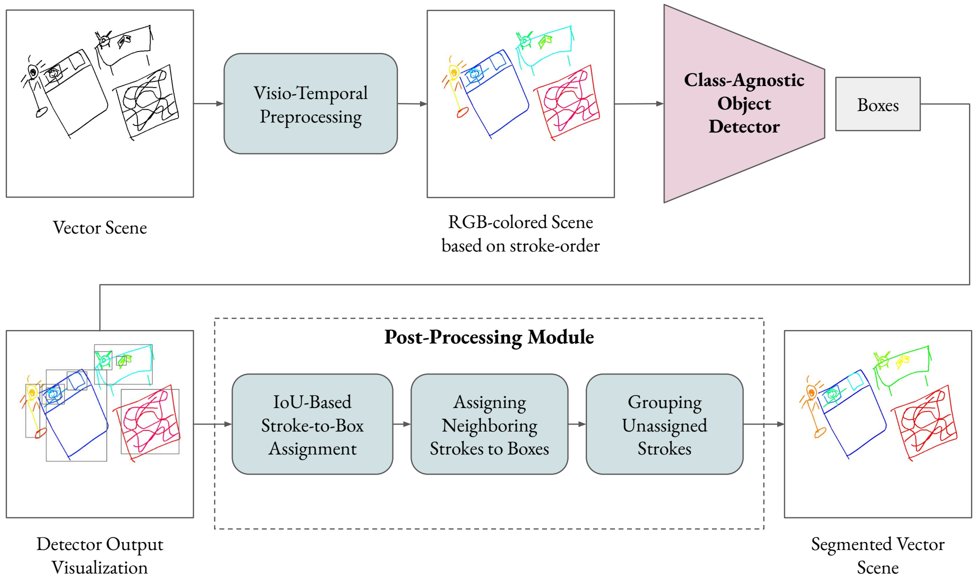

Figure 2. The overall pipeline of CAVT

pixel level. Ge *et al*. [9] proposed a deep-shallow feature fusion network based on DeepLab-v2 [4], examining the influence of local details on scene sketch segmentation. Bourouis *et al*. [2] introduced the first language-supervised scene sketch segmentation method by utilizing sketch captions. In contrast, Zhang *et al*. [32] developed an RNN-GCN-based architecture, marking the first stroke-level approach to scene sketch semantic segmentation. Their study is the most relevant to ours since they also utilize visual, sequential, and spatial information on stroke sequences. However, their code is not publicly available for comparison.

Scene sketch segmentation works mostly focus on assigning each pixel or stroke to a specific class in a given scene. Therefore, different objects belonging to the same category cannot be distinguished at the instance level. In contrast, we propose a novel class-agnostic scene segmentation pipeline that can differentiate object instances in a given scene, regardless of their classes.

2.2. Sketch Datasets

Sketch datasets can be categorized into two primary types: single-object and scene sketch datasets. Single-object sketch datasets feature one object instance per sketch, while scene sketch datasets encompass drawings with multiple objects. Table 1 provides a summary of the sketch datasets and our proposed scene sketch dataset, FrISS.

QMUL Shoe [29], QMUL Chair [29], and Sketchy [20] are multi-modal single-object sketch datasets that contain corresponding natural images paired with each sketch. TU Berlin [7] is the first large-scale free-hand single-object sketch dataset, that is collected via crowdsourcing. QuickDraw [12] is the largest free-hand single sketch dataset, and it is gathered through an online game.

A growing number of large-scale scene-level sketch datasets have been proposed due to the importance of higher-level sketch understanding. SketchyScene [35] pioneered this field, assembling clip art-like single-object sketches onto reference images as layout templates. SketchyCOCO [8] is another synthetically generated scene sketch dataset that integrates free-hand single-object datasets into the corresponding mask area of COCO-Stuff [3] real images. Ge *et al*. [9] introduced two more semi-synthetic scene datasets, called as SKY-Scene and TUB-Scene. Although synthetic scene sketch data generation offers a quick solution to the scarcity of large-scale scene datasets, it lacks the authenticity of human drawing behavior. Moreover, none of these synthetic datasets are available in vector storage formats, rendering them unsuitable for our stroke-based approach. FS-COCO [5] stands out as the first free-hand scene sketch dataset collected in vector format, accompanied by scene captions. However, it lacks stroke- or object-level annotations, hindering semantic segmentation experiments. SFSD [32] is another free-hand scene sketch dataset, offering both vector storage format and stroke-level class annotations, but it is not publicly available. Lastly, CBSC [31] emerges as the sole publicly accessible free-hand scene sketch dataset with instance-level class annotation in the vector storage format. Thus, we leverage CBSC to test our network. To address the lack of free-hand scene sketch datasets, we introduce FrISS, which contains free-hand scene sketches annotated at both instance and stroke levels.

3. Methodology

In this section, the architecture of CAVT and the generation process of its training dataset are explained. As seen in Figure 2, CAVT consists of two sub-modules: (i) the Class-Agnostic Visio-Temporal object detector and (ii) the Post-Processing module. First, each scene sketch is preprocessed using an RGB coloring technique to preserve the temporal stroke order. These color-coded sketches are then passed through the Class-Agnostic Object Detector to generate prediction boxes. Subsequently, the Post-Processing module refines the detector's outputs using a set of rules for stroke-level instance grouping by leveraging temporal stroke order and spatial features. Finally, CAVT produces stroke groups belonging to object instances in the scene.

3.1. Class-Agnostic Visio-Temporal Detector

To proceed with an appropriate object detector, we investigated the cross-domain object detection studies [6, 13, 23] in the literature. DASS-Detector [23] leverages YOLOX [10] and stands out for its high performance within its domain. Inspired by their work, we also utilize YOLOX in our study. Fully-supervised detectors are typically trained to recognize specific predefined classes, restricting their ability to detect objects beyond these predetermined categories. To address this constraint, YOLOX is trained in a class-agnostic manner, in which the detector solely predicts potential object areas without the need for classification. We conduct an ablation study to evaluate the impact of our approach and discuss it in Sec. 5.6. Our trained detector offers predictions concerning potential object regions within sketch scenes. These predictions solely approximate object-bounding boxes on the coordinate plane. Therefore, we introduce a post-processing module designed to group object strokes by leveraging the bounding box predictions.

3.2. Post-Processing Module

This module performs stroke-level segmentation for individual sketches by utilizing the output from the object detector. The full algorithm for the post-processing module is provided in Algorithm 1 in the Supplementary Material. The steps involved in this module are as follows:

1. The predicted bounding boxes are sorted in ascending order based on their area, from smallest to largest.
2. *IoU-Based Stroke-to-Box Assignment:* Starting with the smallest bounding box, the stroke sequence with the highest Intersection over Union (IoU) compared to the selected box is identified. If the IoU value surpasses a threshold called *IoU_threshold*, the corresponding stroke set is assigned to that bounding box.
3. *Assigning Neighboring Strokes to Boxes:* The unassigned strokes are then evaluated based on their overlap ratio. For each of the remaining longest stroke sequences, if the overlap ratio between the sequence and the nearest bounding box exceeds a threshold called *OR_threshold*, the stroke set is assigned to that box. The overlap ratio is calculated by dividing the area of intersection between the bounding box and the stroke set by the total area of the stroke set.
4. *Grouping Unassigned Strokes:* Strokes that remain unassigned to any bounding box after these steps are considered separate objects, and their coordinates are added to the list of predicted boxes.
5. The coordinates of each bounding box are updated based on the latest stroke assignments. Each box's dimensions are adjusted to become the smallest bounding box enclosing its assigned stroke set.
6. These steps are repeated until no further changes occur in stroke groupings, ensuring that each stroke is assigned to a corresponding object bounding box.

Both the *IoU_threshold* and *OR_threshold* are determined using a grid-search algorithm (see in Supplementary Material Sec. S1). The object detector produces bounding boxes without class predictions, so strokes are grouped without class information. This enables the utilization of an external sketch object classifier, offering several advantages: (1) Both stroke- and image-based single sketch classifiers can be employed, each capable of identifying broader or narrower object categories, or sketches with varying complexities; (2) Inference time and required memory can be adjusted based on the chosen classifiers.

3.3. Synthetic Dataset Preparation for Training

Object detection models are widely used in the literature [15, 25, 30]. However, their direct application to the sketch domain faces challenges due to the domain shift from real-life images to scene sketches. Achieving fully supervised detector training on sketches necessitates a large-scale instance-level scene sketch dataset. Furthermore, the training dataset should maintain strokes in vector storage format to utilize temporal cues effectively. Unfortunately, none of the existing large-scale datasets offer both instance-level annotation and vector storage format [5, 8, 9, 35].

To train an object detector for the sketch domain, we created a large-scale, synthetically generated scene sketch dataset. To ensure our object detector's robustness across various categories and drawing styles, we utilized Quick-Draw [12], which offers a wide range of categories and diverse sketch styles. Each scene is composed of a minimum of 2 and a maximum of 8 randomly chosen objects from a pool of 345 categories, with 70K drawing instances per

Dataset	Cats.	Category per sketch			Objects per sketch			Strokes per sketch		
		Max	Min	Mean	Max	Min	Mean	Max	Min	Mean
SketchyScene [35]	45	19	13	6.88	94	3	16.71	-	-	-
SketchyCOCO [8]	17	6	1	2.33	35	2	10.93	-	-	-
CBSC [31]	74	10	3	4.23	16	3	4.72	185	6	33.14
FS-COCO [5]	92 / 150	5 / 25	1 / 1	1.37 / 7.17	-	-	-	561	5	75.86
SFSD [32]	40	11	1	4.46	43	2	7.76	699	9	146.64
FrISS (Ours)	403	10	1	4.33	30	1	6.04	186	4	35.81

Table 2. Comparison and statistics of scene sketch datasets

category. Objects are randomly scaled to have a large side length ranging from 50 to 700 pixels and positioned randomly within the scene. To prevent extreme overlapping between objects, we ensure that the intersection-over-union (IOU) value between them remains below 0.35. Scenes are created in two potential sizes: 720x1280 or 1280x720 pixels. To capture the temporal order, each stroke is assigned a color from a spectrum spanning blue to red based on its order (see Supplementary Material Sec. S2). In total, we generated 11.5K synthetic drawing scenes under these settings, allocating 10K for training and 1.5K for validation.

4. The FrISS Dataset

We propose the largest Free-hand Instance- and Stroke-level Scene sketch dataset (FrISS) that includes scene sketches in vector format, stroke-level class and instance annotations, sketch-text pairs, and verbal audio clips paired with each scene. The data construction process involves two primary stages: (i) sketch collection and (ii) sketch annotation. This section elaborates on these stages and provides statistics and analysis on the FrISS dataset.

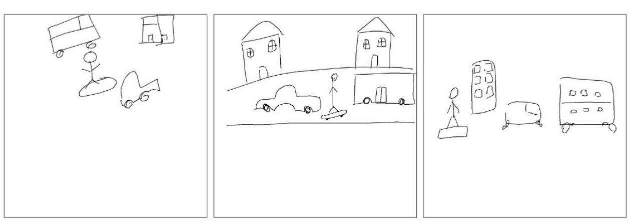

Figure 3. Sample scenes taken from FrISS that are drawn by three individuals by referring to the same textual scene description

4.1. Sketch Collection

We developed a web application to collect scene sketches, following similar data collection methods as in previous studies [5, 32]. Visuals of the web application are provided in Supplementary Material Sec. S5.1. We recruited 100 volunteer participants with varying levels of drawing skills, each tasked with creating 10 distinct scene sketches based on textual scene descriptions. The textual scene descriptions provided during the drawing phase are either sourced from captions within the MS COCO dataset [17] or constructed by us. Details on the generation of scene descriptions, along with examples, are provided in Supplementary Material Sec. S5.3. To avoid influencing participants with predefined layouts or poses, no visual references were provided. As shown in Figure 3, the arrangement and diversity of objects in the scenes varied significantly when participants sketched scenes without visual guidance.

Each participant was given 1.5 minutes to complete each scene. The time limit was determined through pilot studies with a group of volunteers. These studies showed that a shorter time often led to incomplete drawings, while a longer time resulted in excessively detailed sketches. Participants were allowed to redraw objects within the time limit but were not permitted to restart the scene with extra time. Allowing multiple attempts could lead to unrealistically polished sketches. Additionally, participants were asked to verbally describe their scenes as they drew. To ensure comfort and clarity, they were encouraged to speak in their native language. The verbal explanations were recorded during the drawing process, enabling FrISS to support research on tasks such as speech-based sketch studies.

4.2. Sketch Annotation

In the second phase of data collection, participants were presented with scenes they had previously drawn. They annotated each stroke with both instance and category information. Figure 1 shows sample sketch-text pairs from the FrISS dataset and their colored annotations. Different colors are used to visualize instance-level annotations of the objects from the same category (e.g., pizzas, mountains).

To avoid interrupting the natural drawing process, we collected sketch annotations separately from the drawing phase. This phase was conducted under our supervision to ensure accurate annotations. Each stroke in a scene was assigned to its corresponding object category, with incomplete or ambiguous strokes labeled as *'incomplete'* and subsequently excluded from the scene. Additionally, we manually reviewed the annotations for accuracy and assessed the quality of the scenes. Any mislabeled object strokes were either corrected or eliminated from the dataset.

4.3. Statistics and Analysis

Table 2 provides a statistical comparison of various scene sketch datasets, focusing on category, object, and stroke counts per sketch. Our dataset covers a wider range of object categories compared to previous scene datasets. Additionally, each scene sketch was collected within a 1.5-minute timeframe, resulting in simpler sketches resembling participants' daily drawings. Other free-hand scene sketch datasets [5, 32] allow more time for drawings and multiple drawing attempts, which results in extremely detailed scene sketches. On the other hand, our scene sketches contain an average of approximately 36 strokes per scene, significantly fewer than other datasets in terms of complexity. Refer to Figure 1 for sample scenes from our FrISS dataset. Thus, FrISS stands out by including both instance- and stroke-level class annotations. Additional scene samples and comparisons are available in Supplementary Material Sec. S5.

5. Experiments

5.1. Datasets

We utilize temporal stroke information in our pipeline, thus it limits the range of applicable datasets for evaluation. Therefore, we assessed our approach using only the test partitions of FrISS and CBSC [31]. FrISS comprises 1K free-hand scene sketches spanning 403 object categories, with 236 categories overlapping with the QuickDraw classes [12]. We reserved 500 scene sketches for testing, while the remaining sketches were divided into validation (145 sketches) and training (355 sketches) sets. CBSC dataset consists of 222 free-hand scene sketches in its test partition, covering 74 object categories and these categories fully align with QuickDraw, except for the *'person'* class. However, the visual characteristics of the *'yoga'* class of the QuickDraw closely resemble those of the *'person'* class in other scene sketch datasets. Therefore, we map the *'person'* class to *'yoga'* class during the evaluation.

5.2. Sketch Classification

As discussed in Sec. 3, CAVT generates segmented stroke groups without any category assignments. Thus, we utilized one stroke-based and one image-based sketch classifier. First, we investigated the performances of state-of-the-art stroke-based sketch classifiers [11, 19, 26]. Since Sketchformer [19] achieves superior performance, it was selected as the external classifier for categorizing sketches segmented by CAVT. Secondly, we trained various CNN-based classifiers using the training sets of QuickDraw and FrISS. Among these, Inception-V3 [22] outperforms others. Hence, we further utilize our trained Inception-V3 as a second external classifier. In the following sections, we call the end-to-end CAVT + Sketchformer pipeline as *CAVT-S*, and CAVT + pre-trained-Inception-V3 pipeline as *CAVT-I*. A detailed analysis of classifiers can be found in Supplementary Material Sec. S3.

5.3. Evaluation Metrics

Earlier works utilize metrics that are commonly used to evaluate image segmentation models. Hence, we follow the standard four metrics that are used in our competitor models [2, 9, 35] for fair comparison. These metrics are listed as follows: Overall Pixel Accuracy (OVAcc), Mean Pixel Accuracy (MeanAcc), Mean Intersection over Union (MIoU), and Frequency Weighted Intersection over Union (FWIoU). Still, there is no available metric specifically designed for stroke-level scene sketch semantic segmentation. Thus, we propose two additional metrics for stroke-level evaluation:

- **All or Nothing (AoN):** evaluates the ratio of correctly predicted stroke groups. If a single stroke of an object is mislabeled, then the result becomes incorrect.
- **Stroke-level Intersection over Union (S-IoU):** calculates the largest overlap ratio of the actual and the predicted stroke groups, and averages the overlap ratio for all ground truth stroke groups.

Our competitor models perform class-level segmentation and require bitmap images as input. Therefore, we could not compare our results with earlier works on these metrics.

5.4. Implementation Details

The sole trainable component of our network is the Class-Agnostic Visio-Temporal Object Detector, built upon the YOLOX framework [10]. During model training, we employed the MMDetection library, training YOLOX with default configurations while modifying only the total number of categories to 1. Our training process utilizes a single Tesla T4 GPU with a batch size of 16, spanning 600 epochs. We compared our results with the Local Detail Perception (LDP) [9] and the Open Vocabulary (OV) [2]. However, when comparing CAVT with these models, several adjustments to the datasets and our evaluation process are necessary:

- LDP is trained on categories from SKY-Scene [9] and SketchyScene [35]. Additionally, we use Sketchformer [19] as our sketch classifier, which only supports the 345 categories from QuickDraw [12]. To ensure a fair comparison, we created five distinct sub-datasets: *FrISS-SKY* and *CBSC-SKY* include objects from the common classes shared between QuickDraw, SKY-Scene, and FrISS/CBSC; *FrISS-SS* and *CBSC-SS* feature objects from the common categories of

Model	CBSC-SKY				CBSC-SS			
	OVAcc	MeanAcc	MIoU	FWIoU	OVAcc	MeanAcc	MIoU	FWIoU
LDP	54.56	52.82	33.47	37.96	47.85	36.17	23.81	32.93
CAVT-S	70.24	73.89	51.21	59.22	71.25	73.29	51.92	60.30
CAVT-I	**73.76**	**74.08**	**53.38**	**61.89**	**73.13**	**75.26**	**52.45**	**60.56**

Model	FrISS-SKY				FrISS-SS			
	OVAcc	MeanAcc	MIoU	FWIoU	OVAcc	MeanAcc	MIoU	FWIoU
LDP	44.33	27.24	14.91	31.89	41.17	29.97	15.09	27.82
CAVT-S	65.39	**62.33**	**34.88**	**54.86**	60.02	**60.11**	**33.09**	48.11
CAVT-I	**66.56**	62.08	34.18	54.40	**61.54**	55.07	31.83	**48.19**

Table 3. Comparison of CAVT against LDP [9] on the CBSC-SS CBSC-SKY, FrISS-SS, and FrISS-SKY datasets.

Model	CBSC				FrISS-QD				FrISS			
	OVAcc	MeanAcc	MIoU	FWIoU	OVAcc	MeanAcc	MIoU	FWIoU	OVAcc	MeanAcc	MIoU	FWIoU
OV	62.64	62.94	45.15	49.34	64.66	54.67	38.14	50.68	41.13	41.84	25.41	29.92
CAVT-S*	81.21	81.87	68.71	70.13	80.90	**76.99**	64.95	69.53	-	-	-	-
CAVT-I*	**83.52**	**82.36**	**71.97**	**73.14**	**81.89**	75.50	**65.81**	**70.97**	**72.71**	**46.46**	**37.17**	**58.05**

Table 4. Comparison of CAVT with Open Vocabulary (OV) [2] on the CBSC [31], FrISS-QD, and FrISS datasets.

QuickDraw, SketchyScene, and FrISS/CBSC; *FrISS-QD* comprises objects from the common classes of FrISS and QuickDraw.

- The OV model operates without relying on pixel or stroke-level annotations, instead, it uses sketch-caption pairs. During inference, captions are generated by concatenating ground truth object categories, and OV predicts the correct class label from the given set of object classes. To ensure a fair comparison with OV, we developed alternative versions of our pipelines (CAVT-S* and CAVT-I*) that restrict the possible object classes to those present in the ground truth scene.

5.5. Comparison Against State-of-the-art (SOTA)

The comparison results of our model with prior works on the different subsets of CBSC and FrISS datasets are shown in Tables 3 and 4. Across all datasets and metric variations, under identical conditions, the gap between LDP and CAVT-S or CAVT-I is consistently between 15% - 39%, but it narrows to 6% - 31% with OV. Still, our pipeline outperforms previous SOTA by a significant margin.

Figure 4 shows the qualitative comparison between our method, LDP, and OV models. Our pipeline leverages stroke information and does not assign different class labels to any point in a single stroke. This allows us to generate more coherent segmentation outputs. Moreover, we share our instance-level visual results in the rightmost column of the figure. Different from the SOTA models, we can segment different instances from the same category (2nd and 3rd rows). We provide additional visual comparisons in Supplementary Material.

5.6. Ablation Study

In this experiment, we examine the individual effects of each component of CAVT. The key components include the use of temporal stroke order, class-agnostic training, and the post-processing module. To evaluate the impact of the post-processing module, we implement a simple stroke grouping technique as a baseline for comparison. In this method, each stroke is assigned to the nearest predicted bounding box, and the strokes assigned to the same box are grouped as a single object.

Table 5 illustrates the impact of each component on segmentation performance, with each one contributing a notable improvement. While the most significant component in CBSC is PP with a 7.48% average performance increase, CA has the least effect with a 4.96% increase. On the other hand, CA has the most effect on FrISS with an average of 6.22% performance enhancement, while T provides the least increase with 1.82% on average.

As detailed in Section 3.1, the object detector is trained using a synthetic dataset derived from QuickDraw classes [12]. We excluded objects belonging to QuickDraw categories from the FrISS dataset and denoted as FrISS^{sub}. Later, we calculated AoN and S-IoU on this subset to evaluate the generalizability of CAVT to instances from unseen classes. Although the AoN score drops by approximately 10%, the decrease in S-IoU remains only around 1.5%. This indicates that CAVT can still generalize to sketch objects from unseen classes with minimal performance loss.

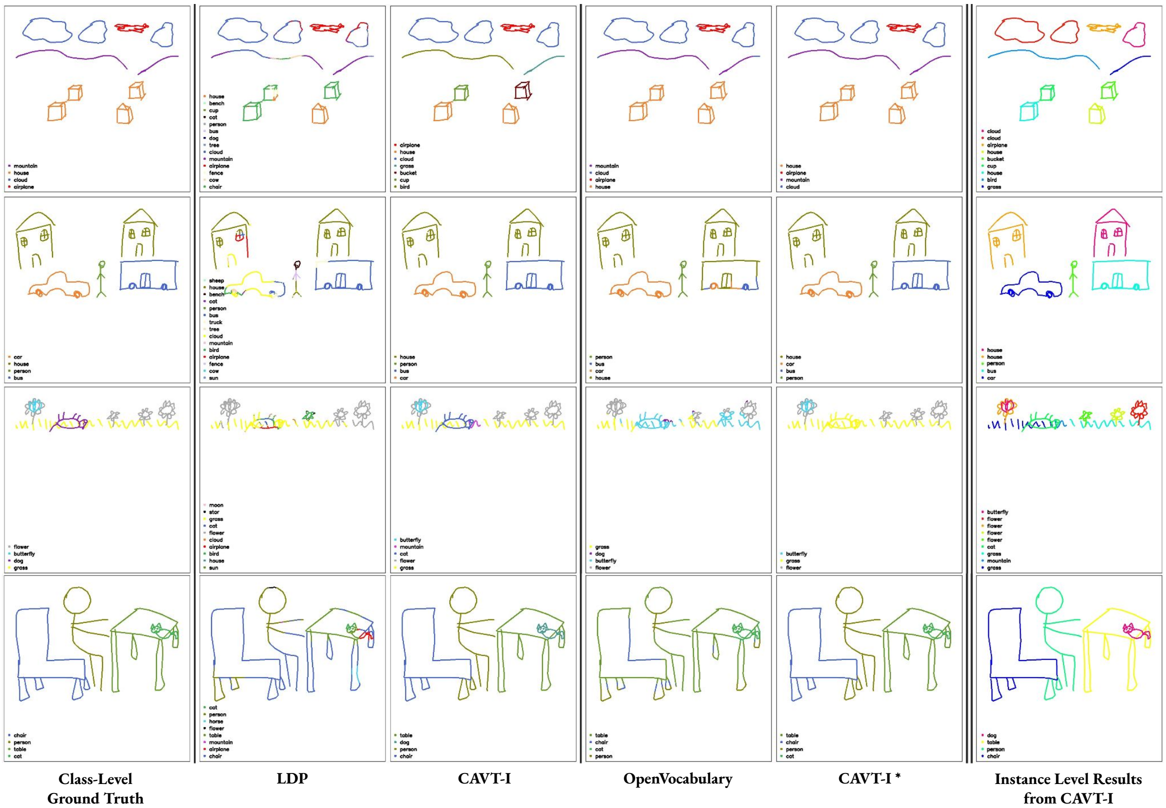

Figure 4. Visual comparison of our method with LDP [9] and OV [2] models that are evaluated on the FrISS-SS dataset.

Components:			CBSC						FrISS						FrISS^{sub}	
T	CA	PP	AoN	S-IoU	OVAcc	MAcc	MIoU	FWIoU	AoN	S-IoU	OVAcc	MAcc	MIoU	FWIoU	AoN	S-IoU
			39.80	68.76	39.54	38.21	23.74	28.53	28.30	57.49	25.08	14.34	6.17	17.32	23.85	58.23
		✓	48.95	73.57	48.83	46.68	30.70	37.05	33.40	60.07	29.77	18.21	8.83	20.74	26.11	58.37
	✓	✓	58.64	81.09	52.00	51.67	32.57	39.55	47.09	**72.89**	32.89	**22.33**	9.98	23.17	39.98	**71.67**
✓	✓	✓	**68.68**	**84.77**	**60.09**	**57.50**	**38.81**	**47.96**	**51.57**	72.77	**36.55**	22.04	**10.20**	**26.12**	**41.62**	71.34

Table 5. Ablation study for the impact of including or excluding three components: temporal stroke order (T), class-agnostic training (CA), and post-processing steps (PP). FrISS^{sub}: calculates metrics for a subset of categories in FrISS that are not part of QuickDraw [12]. The metrics *OVAcc*, *MeanAcc*, *MIoU*, and *FWIoU* are evaluated using CAVT-I, since it also supports the complete class set of FrISS.

6. Conclusion

In this work, we proposed a novel pipeline for the scene sketch semantic segmentation task that identifies individual object instances at both stroke- and instance levels. We utilized both temporal information and the visual appearance of the sketches within a scene. Our approach allows us to assign a class label to each object instance without being constrained by a predefined category list. Furthermore, we introduced the FrISS dataset, comprising instance and stroke-level class annotations, sketch-text pairs, and verbal audio clips paired with each scene. We hope that FrISS facilitates a wide range of studies, including stroke-level scene sketch segmentation, speech-based sketch applications, and cross-modal research utilizing sketch-text pairs. Benefiting from FrISS, we conducted extensive experiments to show that our novel approach outperforms the state-of-the-art methods, yielding more coherent visual results in scene sketch semantic segmentation.

7. Acknowledgements

This study is written as a part of a Research Project supported by grants from the Scientific and Technological Research Council of Turkey (TUBITAK), Turkey (Project No. 120E489). We are also thankful for the support of the council and KUIS AI Center.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 1

[2] Ahmed Bourouis, Judith Ellen Fan, and Yulia Gryaditskaya. Open vocabulary semantic scene sketch understanding. *arXiv preprint arXiv:2312.12463*, 2023. 1, 2, 3, 6, 7, 8

[3] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018. 3

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 1, 3

[5] Pinaki Nath Chowdhury, Aneeshan Sain, Ayan Kumar Bhunia, Tao Xiang, Yulia Gryaditskaya, and Yi-Zhe Song. Fscoco: Towards understanding of freehand sketches of common objects in context. In *European Conference on Computer Vision*, pages 253–270. Springer, 2022. 2, 3, 4, 5, 6

[6] Jinhong Deng, Wen Li, Yuhua Chen, and Lixin Duan. Unbiased mean teacher for cross-domain object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4091–4101, 2021. 4

[7] Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects? *ACM Transactions on graphics (TOG)*, 31(4):1–10, 2012. 2, 3

[8] Chengying Gao, Qi Liu, Qi Xu, Limin Wang, Jianzhuang Liu, and Changqing Zou. Sketchycoco: Image generation from freehand scene sketches. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5174–5183, 2020. 2, 3, 4, 5

[9] Ce Ge, Haifeng Sun, Yi-Zhe Song, Zhanyu Ma, and Jianxin Liao. Exploring local detail perception for scene sketch semantic segmentation. *IEEE Transactions on Image Processing*, 31:1447–1461, 2022. 1, 2, 3, 4, 6, 7, 8

[10] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 4, 6

[11] David Ha and Douglas Eck. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*, 2017. 6

[12] David Ha and Douglas Eck. A neural representation of sketch drawings. In *International Conference on Learning Representations*, 2018. 2, 3, 4, 6, 7, 8

[13] Junguang Jiang, Baixu Chen, Jianmin Wang, and Mingsheng Long. Decoupled adaptation for cross-domain object detection. *arXiv preprint arXiv:2110.02578*, 2021. 4

[14] Kurmanbek Kaiyrbekov and Metin Sezgin. Deep strokebased sketched symbol reconstruction and segmentation. *IEEE computer graphics and applications*, 40(1):112–126, 2019. 1, 2

[15] Chuyi Li, Lulu Li, Yifei Geng, Hongliang Jiang, Meng Cheng, Bo Zhang, Zaidan Ke, Xiaoming Xu, and Xiangxiang Chu. Yolov6 v3. 0: A full-scale reloading. *arXiv preprint arXiv:2301.05586*, 2023. 4

[16] Lei Li, Hongbo Fu, and Chiew-Lan Tai. Fast sketch segmentation and labeling with deep learning. *IEEE computer graphics and applications*, 39(2):38–51, 2018. 1, 2

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 5

[18] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015. 1

[19] Leo Sampaio Ferraz Ribeiro, Tu Bui, John Collomosse, and Moacir Ponti. Sketchformer: Transformer-based representation for sketched structure. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14153–14162, 2020. 6

[20] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics (TOG)*, 35(4):1–12, 2016. 2, 3

[21] Zhenbang Sun, Changhu Wang, Liqing Zhang, and Lei Zhang. Free hand-drawn sketch segmentation. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part I 12*, pages 626–639. Springer, 2012. 1

[22] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 6

[23] Barış Batuhan Topal, Deniz Yuret, and Tevfik Metin Sezgin. Domain-adaptive self-supervised pre-training for face & body detection in drawings, 2023. 4

[24] Xingyuan Wu, Yonggang Qi, Jun Liu, and Jie Yang. Sketchsegnet: A rnn model for labeling sketch strokes. In *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2018. 1, 2

[25] Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu. End-to-end semi-supervised object detection with soft teacher. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3060–3069, 2021. 4

[26] Peng Xu, Chaitanya K Joshi, and Xavier Bresson. Multigraph transformer for free-hand sketch recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 33(10):5150–5161, 2021. 6

[27] Jie Yang, Aihua Ke, Yaoxiang Yu, and Bo Cai. Scene sketch semantic segmentation with hierarchical transformer. *Knowledge-Based Systems*, 280:110962, 2023. 1, 2

[28] Lumin Yang, Jiajie Zhuang, Hongbo Fu, Xiangzhi Wei, Kun Zhou, and Youyi Zheng. Sketchgnn: Semantic sketch seg-

mentation with graph neural networks. *ACM Transactions on Graphics (TOG)*, 40(3):1–13, 2021. 1, 2

[29] Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, and Chen-Change Loy. Sketch me that shoe. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 799–807, 2016. 2, 3

[30] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 4

[31] Jianhui Zhang, Yilan Chen, Lei Li, Hongbo Fu, and Chiew-Lan Tai. Context-based sketch classification. In *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch-Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering*, pages 1–10, 2018. 2, 3, 5, 6, 7

[32] Zhengming Zhang, Xiaoming Deng, Jinyao Li, Yukun Lai, Cuixia Ma, Yongjin Liu, and Hongan Wang. Stroke-based semantic segmentation for scene-level free-hand sketches. *The Visual Computer*, 39(12):6309–6321, 2023. 1, 2, 3, 5, 6

[33] Yixiao Zheng, Jiyang Xie, Aneeshan Sain, Yi-Zhe Song, and Zhanyu Ma. Sketch-segformer: Transformer-based segmentation for figurative and creative sketches. *IEEE transactions on image processing*, 2023. 1, 2

[34] Xianyi Zhu, Yi Xiao, and Yan Zheng. Part-level sketch segmentation and labeling using dual-cnn. In *Neural Information Processing: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13-16, 2018, Proceedings, Part I 25*, pages 374–384. Springer, 2018. 1, 2

[35] Changqing Zou, Qian Yu, Ruofei Du, Haoran Mo, Yi-Zhe Song, Tao Xiang, Chengying Gao, Baoquan Chen, and Hao Zhang. Sketchyscene: Richly-annotated scene sketches. In *ECCV*, pages 438–454. Springer International Publishing, 2018. 1, 2, 3, 4, 5, 6