Sketch-Vision: Primitive Detection and Program Reconstruction

D1.2 Progress Report

Team JAXAXAX

Nikita Zagainov (n.zagainov@innopolis.university)
Nikita Tsukanov (n.tsukanov@innopolis.university)
Said Kadirov (s.kadirov@innopolis.university)
Tetkin Dmitry (d.tetkin@innopolis.university)

November 24, 2025

Abstract

We extend raster-to-program modeling to hand-drawn sketches with engineering annotations. This interim deliverable documents (i) a lightweight synthetic dataset generator with JSON annotations, (ii) visualization and evaluation utilities, and (iii) updated repository documentation. We outline alignment with CAD2Program-style VLMs and next steps.

1 Repository

https://github.com/V1adych/sketch-vision-pmldl

2 Overview

Our goal is to detect digits, arrows, dimensions, radii, and geometric primitives in noisy sketches, producing a structured representation suitable for downstream CAD. We follow the encoder–decoder paradigm (ViT encoder + LM decoder) summarized in our prior notes (see `docs/sketch-vision-eng.pd`

3 Repository Updates in D1.2

3.1 Synthetic Dataset Generator

We added `preprocessing/generate_synthetic.py` that renders rectangles, circles, and line segments with basic dimension labels and produces aligned JSON annotations. It also writes train/val/test splits.

3.2 Visualization and Evaluation

`preprocessing/visualize_annotations.py` overlays bounding boxes and labels on top of images. `evaluation/metrics.py` provides IoU; `evaluation/evaluate_synthetic.py` prints simple corpus stats per split.

3.3 Documentation

`README.md` now includes a Quickstart showing how to generate a small dataset, visualize a sample, and get stats.

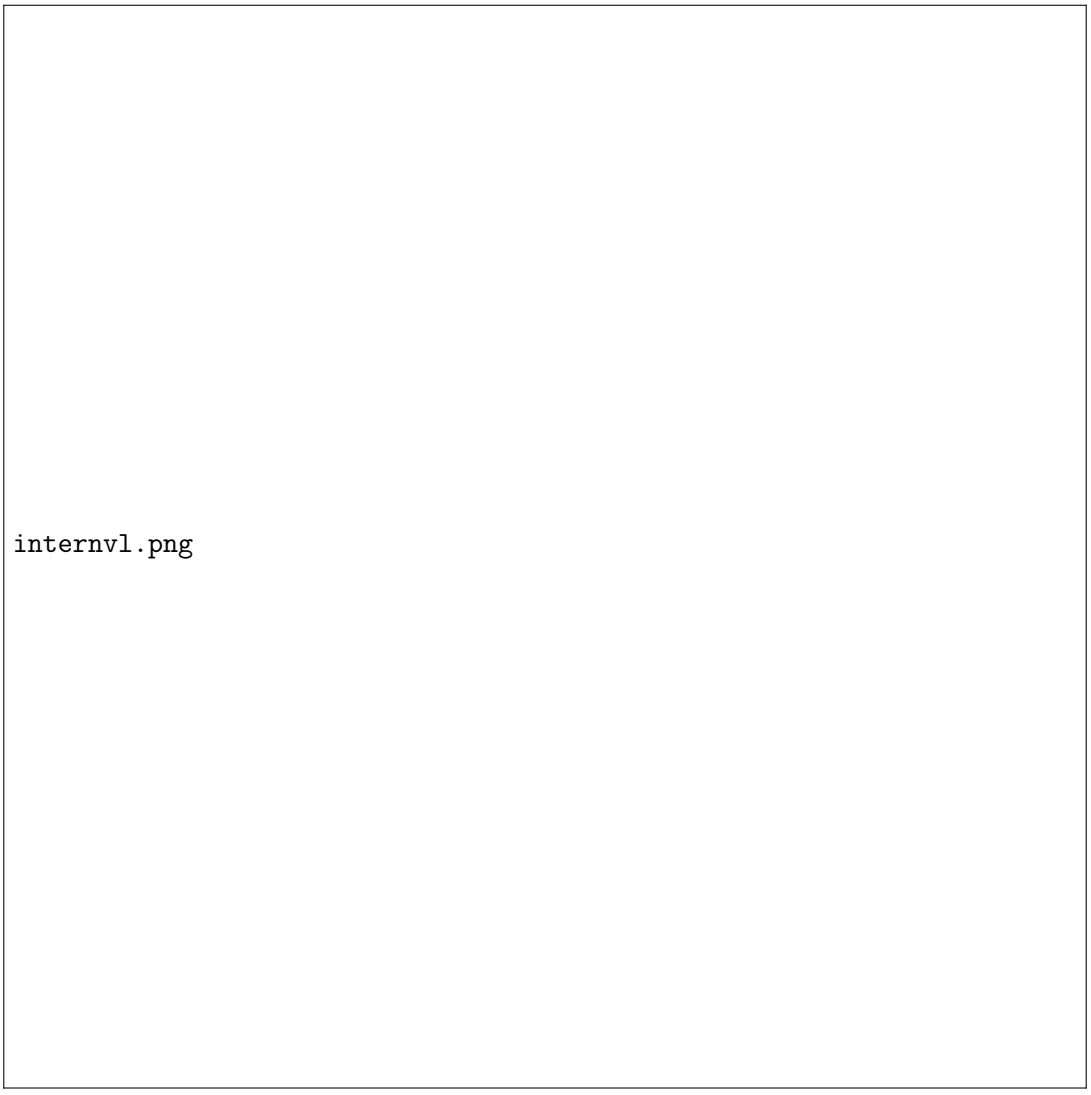

Figure 1: High-level multimodal architecture (reference figure in repo).

4 Quickstart (Reproducibility)

```
python preprocessing/generate_synthetic.py --output-dir dataset/synthetic --num-samples 100
python preprocessing/visualize_annotations.py \
  --images-dir dataset/synthetic/images \
  --annotations-dir dataset/synthetic/annotations \
  --name sketch_00010 --output dataset/synthetic/vis
python evaluation/evaluate_synthetic.py \
  --annotations-dir dataset/synthetic/annotations \
  --splits dataset/synthetic/splits/train.txt

# New in this update:
python preprocessing/extract_text_tokens.py \
  --images-dir dataset/synthetic/images \
  --annotations-dir dataset/synthetic/annotations
python scripts/train_detector.py --data-root dataset/synthetic --epochs 1 --batch-size 2
python evaluation/sequence_metrics.py \
  --annotations-dir dataset/synthetic/annotations \
  --splits dataset/synthetic/splits/test.txt \
  --predictions predictions.tsv
```

Generated visualizations can be included in the appendix; see `docs/example.png` for style reference.

5 Data Visualization

We include dataset summary plots generated from the synthetic annotations:

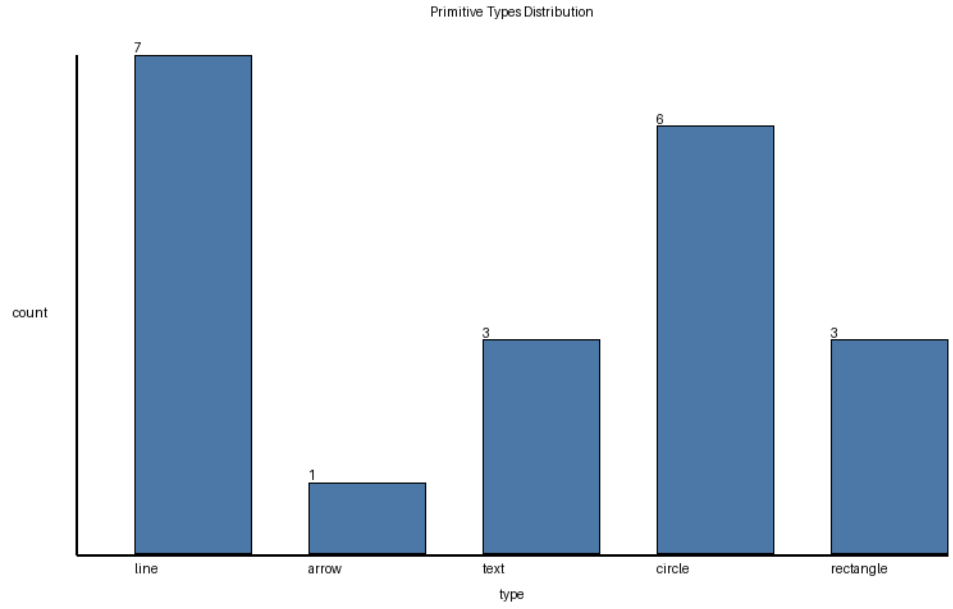

Figure 2: Distribution of primitive types in the split.

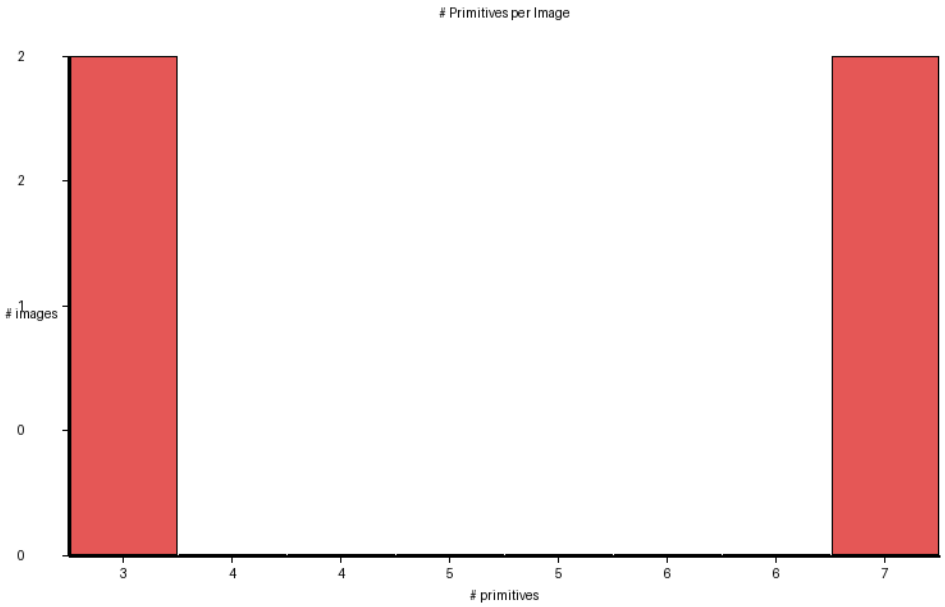

Figure 3: Per-image primitive counts.

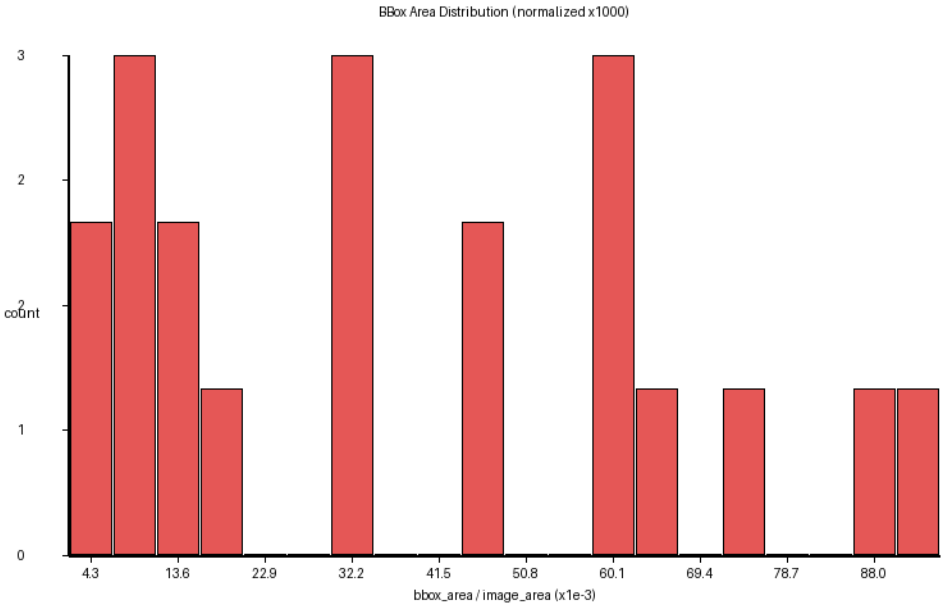

Figure 4: Bounding-box area distribution (normalized by image area).

6 Implemented Updates aligned with Planned Work

- **Expanded primitives**: Added `arrow` and `text` tokens to the synthetic generator, updated visualization.
- **OCR integration**: Added a utility to extract text tokens via Tesseract and store them next to annotations.
- **Detector baseline**: Provided a minimal training script (Faster R-CNN via torchvision) to train on synthetic data.
- **Metrics**: Added precision/recall/F1 utilities and sequence-level accuracy (exact match, char accuracy).
- **Program serialization**: Synthetic annotations now include a `program` string for encoder–decoder evaluation.

7 Planned Work

- Expand primitive set (arrows, dimensions-as-text tokens) and OCR integration.
- Train a detector baseline (ViT/DETR) on synthetic data; add metrics (precision/recall/F1).
- Integrate encoder–decoder path for program reconstruction; evaluate sequence accuracy.