Segment-Then-Classify: Few-shot instance segmentation for environmental remote sensing

DISCLAIMER: Summarized by AI

Problem they are trying to solve / Purpose of method

Environmental remote sensing tasks like land cover classification and glacier monitoring depend on instance segmentation, but current methods like YOLOv8 require large amounts of annotated data, which is difficult to obtain in environmental sciences.

The paper introduces a method to:

- Solve the **data scarcity issue** in instance segmentation.
- Enable high performance with **few labeled examples**.
- Reduce the **need for labor-intensive annotation** and reliance on models not optimized for remote sensing.

How does it differ from other methods?

Unlike traditional **Detect-then-Segment** models that:

- Detect objects using bounding boxes before segmentation,
- Require extensive training data and manual labels,
- Struggle with remote sensing applications,

The proposed **Segment-then-Classify (STC)** approach:

- Uses **SAM (Segment Anything Model)** in zero-shot mode to generate all possible object masks without prior training,
- Then classifies the segmented objects using a **lightweight image classifier** like ViT (Vision Transformer),
- Achieves comparable or better accuracy with **just 1/3 of the data** needed by YOLOv8.

How the method works

Overview (Simple)

1. **Segment:** Use SAM in "everything" mode to segment all objects in an image without training.
2. **Classify:** Use a trained classifier (ViT) to identify and filter relevant objects among the segmented outputs.

Detailed Workflow

1. **Segmentation (SAM):**
    - Input image is divided using a 32x32 point grid.

- Each point acts as a prompt for object mask generation.
- Filters applied to clean noise and remove overlapping masks using Non-Maximum Suppression.

2. **Classification (ViT):**
   - Image patches are cropped to each mask's bounding box.
   - Trained ViT assigns each patch to one of 10 target classes (plus an "other" class for irrelevant items).
3. **Training & Evaluation:**
   - Evaluated on the **NWPU VHR-10 dataset** with 10 object categories.
   - mAP@0.5 used to assess performance.

Key Results

- **High Data Efficiency:** ViT converged with just 40 training examples per class.
- **Performance Comparison:** STC outperforms YOLOv8 with significantly less training data.
- **Top Category Performance:** Up to 0.99 mAP@0.5 for classes like "Storage Tanks".
- **Challenges:** Lower performance in cluttered or small-object classes (e.g., "Bridges" at 0.07 mAP), suggesting SAM may need fine-tuning for certain use cases.