

Run Commands

1. **preprocessing.py** — preprocessing and test data augmentation

cd Code – to start from the “Code” folder

```
python preprocessing.py
```

2. **preprocessing_expanded.py** – expanded preprocessing (CLIP cosine similarity and SAM2 crop)

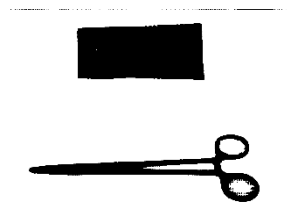
cd Code – to start from the “Code” folder

```
python preprocessing_expanded.py `
--checkpoint sam_vit_h_4b8939.pth `
--model_type vit_h `
--classes scissors `
--threshold 0.3 `
--device cpu
```



3. **sam2_mask_generator.py** — mask generation with CLIP filtering
Start from the “Test 3 - SAM2 Mask Generator - M” folder.

```
python sam2_mask_generator.py `
--image IMG_0851.jpeg `
--checkpoint sam_vit_h_4b8939.pth `
--model_type vit_h `
--output_dir masks_output `
--threshold 0.3 `
--visualize
```



4. **autolabel_and_train.py** — automatic pseudo-labeling and training initiation

Start from the “Test 4 - Autolabel and Train” folder.

a. Auto-labeling masks:

```
python autolabel_and_train.py autolabel `
--masks_dir sam_output/ `
--images_dir dataset/images/ `
--output_dir dataset/masks/ `
--classes      right_angle_clamp      curved_mosquito      DeBakey_forceps
angled_bulldog_clamp `
--threshold 0.3 `
--device cpu
```

b. Training model:

```
python autolabel_and_train.py train `
--dataset_root dataset/ `
--classes      right_angle_clamp      curved_mosquito      DeBakey_forceps
angled_bulldog_clamp `
--epochs 15 `
--batch_size 4 `
--lr 0.005 `
--device cpu
```

```
Epoch 1/15: 100%|████████████████████████████████████████████████████████████████████████████████| 3/3 [01:23<00:00, 27.92s/it]
Epoch 1 total loss: 5.6356
Epoch 2/15: 100%|████████████████████████████████████████████████████████████████████████████████| 3/3 [01:17<00:00, 25.73s/it]
Epoch 2 total loss: 2.3859
Epoch 3/15: 100%|████████████████████████████████████████████████████████████████████████████████| 3/3 [02:08<00:00, 42.75s/it]
Epoch 3 total loss: 0.4022
Epoch 4/15: 100%|████████████████████████████████████████████████████████████████████████████████| 3/3 [01:55<00:00, 38.63s/it]
Epoch 4 total loss: 0.0014
```

5. **classified_crops.py** – automated object detection and classification

Start from the “Test 5 - Classified Crops - M” folder.

```
python classified_crops.py
```



6. sam2_surgical_mask.py – precise surgical instrument mask generation

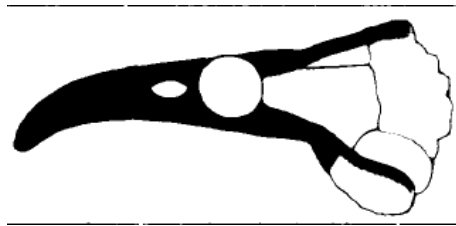
Start from the “Test 6 - SAM2 Surgical Mask - M” folder.

```
python sam2_surgical_mask.py `
--input images/surgery_scene.jpeg `
--output masks/surgical_mask.png `
--model-type vit_h `
--checkpoint sam_vit_h_4b8939.pth `
--device cpu `
--method largest
```

**7. generate_masks_with_sam.py** – binary segmentation masks

Start from the “Test 7 - Generate Masks with SAM - M” folder.

```
python generate_masks_with_sam.py
```

**8. Model LoRA 1:**

- a. **inference.py** – Inference and Multi-Object Detection Using SAM2 + ViT+LoRA
- b. **model.py** – Fine-Tuning ViT with LoRA for Instrument Classification

9. Model LoRA 2:

- a. **inference.py** – semantic segmentation
- b. **surgical_instrument_segmentation.py** – semantic segmentation model (SegFormer with LoRA)

10. Model YOLO LoRA – YOLOv8 Custom Training Pipeline