AdipoSAM Setup and Usage Guide

Author: michael.b.anderson@okstate.edu

Overview

AdipoSAM is a Python-based image analysis tool utilizing Meta AI's Segment Anything Model (SAM) for automated segmentation and measurement of adipocytes in microscopy images. It offers customization of detection parameters, filtering options, and outputs detailed measurements and visualizations.

System Requirements

- Python 3.x
- pip
- Git
- Internet access for package installation
- Compatible with Windows, macOS, and Linux

Model Setup

Due to the large file size of the SAM model checkpoint (sam_vit_b_01ec64.pth), it is not included in the AdipoSAM package. Users must manually download the model from the official Meta AI repository at:
https://github.com/facebookresearch/segment-anything

After downloading sam_vit_b_01ec64.pth, place the file in the root directory of the AdipoSAM folder or specify the path to it within the sam_processor.py script under the model loading section. The script will look for this file during initialization, so ensure it is correctly referenced or located in the expected directory.

Folder Hierarchy

```
├──Input
├──Output
└──Utilities
   └──AdipoSAM - previous versions
      ├──1 to 38 (various developmental versions)
```

Setup Instructions

1. Open Command Prompt (Windows) or Terminal (Mac/Linux):

   - Windows: Press Windows Key, type 'cmd', and press Enter.
   - Mac: Press Command + Space, type 'Terminal', and press Enter.
   - Linux: Use Ctrl + Alt + T or search for 'Terminal'.

2. Navigate to Your Project Folder:

cd path/to/segment_anything

3. (Optional) Create and Activate a Virtual Environment:

python -m venv sam-env
sam-env\Scripts\activate (Windows)
source sam-env/bin/activate (Mac/Linux)

4. Install Required Python Packages:

pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cpu
pip install opencv-python numpy pandas scipy matplotlib

5. Clone and Install Segment Anything:

git clone https://github.com/facebookresearch/segment-anything.git
cd segment-anything
pip install -e .

6. ⚠ Critical Path Configuration

a) open AdipoSAM-v0.40.py in Visual Studio Code (free Microsoft software)

b) update line 122 with the path for your computer

sam_checkpoint = "C:/Your/Path/To/ AdipoSAM/sam_vit_b_01ec64.pth"

c) update lines 209 and 210 with the path for your computer

input_folder = "C:/ Your/Path/To /AdipoSAM/Input"

output_folder = "C:/ Your/Path/To /AdipoSAM/Output"

note: These paths are hard-coded and must be modified to match your local file system. Failure to update them will result in file not found or write errors.

7. Prepare Your Image Files:

Place all input images (.png, .jpg, .jpeg) into the 'Input' folder.

8. Run the Segmentation Script:

python AdipoSAM-v0.40.py

Output Description

- Binary Masks (.tiff) → Output/binary_masks/
- Segmented Overlay Images (.png) → Output/segmented_outputs/

- Individual CSV Files → Output/data/
- Summary CSV File → Output/

Adjustable Parameters

Located within the script functions such as:
- setup_sam()
- filter_masks()
- analyze_cell_properties()
- save_results()

Usage Tips

- Ensure correct Python dependencies
- Adjust pixel_to_micron_ratio and filtering parameters as needed
- Review visual and numerical outputs in the 'Output' directory

Citations and Acknowledgments

Please cite the appropriate papers and acknowledge the use of AdipoSAM and the Segment Anything Model in any published work.

For inquiries: michael.b.anderson@okstate.edu

SAM Cell Analysis Script Documentation

Overview

This script uses the Segment Anything Model (SAM) to detect, analyze, and measure cells in microscopy images. It includes features for excluding cells that touch image borders and provides detailed measurements of cell properties.

Required Dependencies

```
# Core Dependencies
opencv-python>=4.8.0
numpy>=1.24.0
torch>=2.0.0
pandas>=2.0.0
scipy>=1.10.0

# SAM-specific Dependencies
segment-anything @ git+https://github.com/facebookresearch/segment-anything.git
```

Directory Structure

```
C:\Users\AeonS\Desktop\segment_anything\
├── input\                    # Input Directory
│   ├── image1.png           # Support formats: .png, .jpg, .jpeg
│   └── image2.png
│
├── sam_vit_b_01ec64.pth     # SAM Model Checkpoint
│
└── output\                   # Output Directory
    ├── summary.csv          # Combined measurements from all images
    │
    ├── data\                # Individual Measurements
    │   ├── image1_cell_measurements.csv
    │   └── image2_cell_measurements.csv
    │
    ├── binary_masks\        # Binary Mask Files
    │   ├── image1_binary_mask.tiff
    │   └── image2_binary_mask.tiff
    │
    └── segmented_outputs\   # Visualization Outputs
        ├── image1_segmented_output.png
        └── image2_segmented_output.png
```

Adjustable Parameters

1. Cell Detection and Filtering Parameters

Location: filter_masks() function

```
filter_masks(
    min_circularity=0.64,      # Cell roundness (0-1)
    min_area=100,              # Minimum cell area (sq microns)
    max_area=30000,            # Maximum cell area (sq microns)
    max_aspect_ratio=12,       # Maximum length/width ratio
    min_intensity=0.65,        # Minimum brightness threshold (0-1)
    border_margin=10           # Exclusion margin from image edge (pixels)
)
```

2. Cell Measurement Parameters

Location: analyze_cell_properties() function

```
analyze_cell_properties(
    pixel_to_micron_ratio=0.4942  # Conversion factor for measurements
)
```

3. SAM Model Configuration

Location: setup_sam() function

```
SamAutomaticMaskGenerator(
    points_per_side=32,                    # Detection density
    pred_iou_thresh=0.88,                  # Prediction confidence threshold
    stability_score_thresh=0.92,           # Mask stability threshold
    crop_n_layers=1,                       # Number of crop layers
    crop_n_points_downscale_factor=2,      # Downscale factor for crop points
    min_mask_region_area=50                # Minimum region size
)
```

4. Border Check Parameters

Location: touches_border() function

```
touches_border(
    border_margin=1  # Margin for border check (pixels)
)
```

5. Output Visualization Parameters

Location: save_results() function

```
cv2.addWeighted(
    image, 0.7,     # Original image weight
    vis_mask, 0.3,  # Overlay mask weight
    0               # Gamma correction
)
```

Output Files

1. Individual Measurement CSVs

Location: output/data/ Contents:

- filename
- area_sq_microns
- circularity
- aspect_ratio

2. Summary CSV

Location: output/

- Combines all individual measurements
- Maintains same columns as individual CSVs

3. Visual Outputs

- Binary Masks (TIFF format)
    - Location: output/binary_masks/
    - Pure binary representation of detected cells
- Segmented Outputs (PNG format)
    - Location: output/segmented_outputs/
    - Original image with cell outlines overlaid
    - Green contours on original image

Key Functions

Processing Functions

```
process_image()      # Main image processing pipeline
filter_masks()       # Applies all filtering criteria
analyze_cell_properties()  # Calculates cell measurements
```

File Management Functions

```
export_csv()         # Saves individual measurements
combine_csv_files() # Creates summary CSV
save_results()       # Handles all file outputs
```

Usage Tips

1. Ensure all dependencies are installed at the correct versions
2. Place input images in the input directory
3. Configure parameters based on your specific needs:
    - Adjust pixel_to_micron_ratio for your microscope calibration
    - Modify min_area and max_area based on expected cell sizes
    - Tune border_margin based on image edge quality
4. Run the script to process all images in the input directory
5. Check the output directory for results

Performance Considerations

- GPU acceleration is available if CUDA is detected
- Processing time scales with image size and cell count
- Memory usage depends on image dimensions and number of detected cells