EXTRACTING ROI COORDINATES FROM VIDEO FEEDS

Documentation for “get_ROI.py”

.

TLDR:

- Extract coordinates from videos using this script.

  Steps:

  1. Configure file path in the code itself
  2. Run the script
  3. Pause the video using ‘p’
  4. Click on the required points to mark it
  5. Form the Region of Interest (ROI) Polygon
  6. Press ‘q’ to quit
  7. Coordinates are returned to the terminal

- One place to be careful is while selecting the coordinates for the polygon, it is not necessary to close the polygon as 4 coordinates have already been selected. If the polygon is closed, then ignore the 5th coordinate that gets printed on the terminal.
- Make sure the size of the processing video and the “target_size” variable in this script are same, if not polgon will get misplaced.
- NOTE: This runner code “app.py” already resizes the video to a standard 640px x 640px size. Which is set to be the default in the “get_ROI.py” script.

Overview

This script is designed to select a Region of Interest (ROI) in a video feed, often for tasks like object detection or tracking. The script allows users to draw polygons on individual frames of the video to select an ROI. The selected coordinates of the ROI are displayed in the terminal, allowing for easy integration into larger projects. This tool is particularly useful in scenarios where manual selection of specific areas in a video is required for further analysis or processing.

Dependencies

- cv2: OpenCV library for computer vision tasks.
- numpy: A fundamental package for numerical computations in Python.

Functionality

The script provides an interactive interface to manually select a polygonal ROI in a video frame using mouse clicks. The ROI is selected by clicking on the video frame, and the coordinates of the selected polygon are returned and displayed in the terminal.

Script Components

1. Global Variables

- video_path: Path to the video file from which ROI coordinates will be extracted.
- roi_points: List to store the coordinates of the selected polygon vertices.
- drawing: Boolean flag to indicate whether the user is currently drawing a polygon.
- current_frame: Holds the current frame of the video being displayed and processed.
- target_size: Tuple specifying the target size to which each frame will be resized while maintaining the aspect ratio.

2. draw_polygon Function

- **Purpose**: Handles mouse events to allow the user to draw a polygon on the video frame by clicking on it.
- **Arguments**:
  - event: The type of mouse event (e.g., left button click, right button click).

- x, y: Coordinates of the mouse pointer when the event occurs.
- flags, param: Additional parameters required by OpenCV's setMouseCallback.

- **Process**:
  - On EVENT_LBUTTONDOWN: Starts drawing the polygon and stores the clicked point in roi_points. A small circle is drawn at the clicked point, and a line is drawn connecting consecutive points.
  - On EVENT_RBUTTONDOWN: Completes the polygon by drawing a line connecting the last point to the first.

3. resize_with_padding Function

- **Purpose**: Resizes the video frame to the specified target size while maintaining the original aspect ratio. Pads the frame with black borders if necessary to fit the target dimensions.
- **Arguments**:
  - image: The input video frame to be resized.
  - target_size: The desired size of the resized frame.
- **Returns**: A resized frame with padding as needed to maintain the aspect ratio.

4. select_roi Function

- **Purpose**: Captures frames from the video, allows the user to draw a polygonal ROI on the frames, and returns the coordinates of the ROI.
- **Arguments**:
  - video_path: Path to the video file from which the frames are captured.
- **Process**:
  - Opens the video file and sets up a window for displaying frames.
  - Registers the draw_polygon function as the callback for mouse events.
  - Allows the user to pause/play the video with the 'p' key and quit with the 'q' key.
  - As frames are played, the user can draw polygons on them, and the coordinates of the drawn polygon are captured.
- **Returns**: An array of coordinates representing the vertices of the selected polygon.

5. Script Execution

- The script captures ROI points by calling the select_roi function.
- The selected ROI coordinates are printed in the terminal.

Usage

1. **Execution**:
   - Run the script using “python get_ROI.py”.
   - The video specified by video_path will start playing in a window.
2. **Controls**:
   - **Play/Pause**: Press 'p' to pause and resume the video.
   - **Quit**: Press 'q' to quit the video window.
   - **Drawing the Polygon**:
     - **Left Click**: Start drawing the polygon. Each click adds a vertex to the polygon.
3. **Output**:
   - The coordinates of the selected ROI polygon are displayed in the terminal.

Example Output

Selected ROI coordinates: [[100 150]

[200 250]

[300 350]

[100 350]]

Notes for Developers

- **Extensibility**: The script can be easily integrated into larger computer vision projects where ROI selection is required.
- **Customization**: The video path, target size, and other parameters can be customized to fit specific use cases.
- **No File Saving**: The script does not save the ROI coordinates to a file; it only displays them in the terminal for quick reference. Coordinates are hence needed to be copied to the “camera_profiles.xlsx” in the suitable format for further usage.