Real-Time Object Tracking Simulation

Author: David Müller

Documentation

This documentation should guide users on how to effectively use the object detector, the tracker, and run the app. Users can customize parameters according to their specific requirements.

1. Object Detector

Class: ShapeDetector

Description:

- The object detector is implemented through the ShapeDetector class. It identifies shapes, specifically circles and squares, within images.

Methods

Initialization:

- The __init__ method initializes the ShapeDetector object with parameters required for shape detection.

Parameters:

- hough_params: Parameters for Hough circle detection algorithm.
- threshold_value: Threshold value for binary thresholding.
- max_value: Maximum value for binary thresholding.
- threshold_type: Type of thresholding method.

Detection:

- The *detect_shapes* method takes an image as input and returns a list of detected shapes. Each shape is represented by a tuple containing its coordinates, dimensions and type.

2. Tracker

Class: Tracker

The tracker, implemented through the Tracker class, is responsible for detecting shapes in a video file and tracking their positions over time.

Attributes:

- video_path: Path to the input video file.
- shape_detector: An instance of the ShapeDetector class used for detecting shapes in each frame of the video.

Methods:

Initialization:

- The __init__ method initializes the Tracker object with the video file path and detection parameters.

Parameters:

- video_path: Path to the input video file.
- hough_params: Parameters for Hough circle detection algorithm.
- threshold_value: Threshold value for binary thresholding.
- max_value: Maximum value for binary thresholding.
- threshold_type: Type of thresholding method.

Shape detection and tracking in the video:

- The *detect_shapes_in_video* method detects shapes in the input video and visualizes their positions over time. It processes each frame, detects shapes, and visualizes them with bounding boxes and a path.

3. Running the App

1. Install the necessary libraries by running pip install opencv-python.
2. Download the provided script (object_detection_and_tracking.py) and the video file (luxonis_task_video.mp4) to the same directory.
3. Run the script object_detection_and_tracking.py.
4. The application will open a window displaying the video feed with bounding boxes around detected squares.
5. Press q to exit the application.