To execute the script, simply run **worker\_ocr.py**.

**Here's the workflow breakdown:**

* **Object Detection**: YOLOv5 identifies shop names' locations, enabling extraction of specific regions.
* **Region Extraction**: Utilizing YOLOv5's output coordinates, we isolate the shop name areas.
* **OCR Detection**: PaddleOCR analyzes cropped images, discerning text within them.
* **Timestamping**: Initial and final timestamps for each shop name appearance are generated based on a logical inference.

**Challenges:**

* **Unpredictable Text Visibility**: As the video is a blog, text may not always appear as anticipated. Partial visibility during movement complicates detection.
* **Distant Shop Names**: If shop names are too far or unclear, both object detection and OCR performance may suffer.
* **OCR Accuracy**: Inaccurate predictions sometimes occur, particularly in challenging conditions.

**To address these challenges effectively, consider the following refined approach:**

* **Prioritize Fully Visible Shop Names**: Emphasize the extraction of fully visible shop names to enhance accuracy. This can be achieved by setting thresholds for visibility based on factors such as size and clarity of text.
* **Handling Partially Visible Texts**: Develop a logic to handle partially visible texts. Only append shop names if they match predefined criteria or are present in a specified list. By implementing this logic, mispredictions due to partial visibility can be mitigated effectively.
* **Dealing with Distant Shop Names**: Create a curated list of distant shop names based on known locations or characteristics. Utilize this list to refine predictions for distant shop names and reduce inaccuracies caused by distance.
* **Continuous Improvement**: Regularly update and refine the predefined criteria, predefined lists, and curated lists based on feedback and real-world observations. This iterative approach ensures adaptability and improves the accuracy of the system over time.