Crowd Monitoring & QA

[Week 7]

This Week's Goals

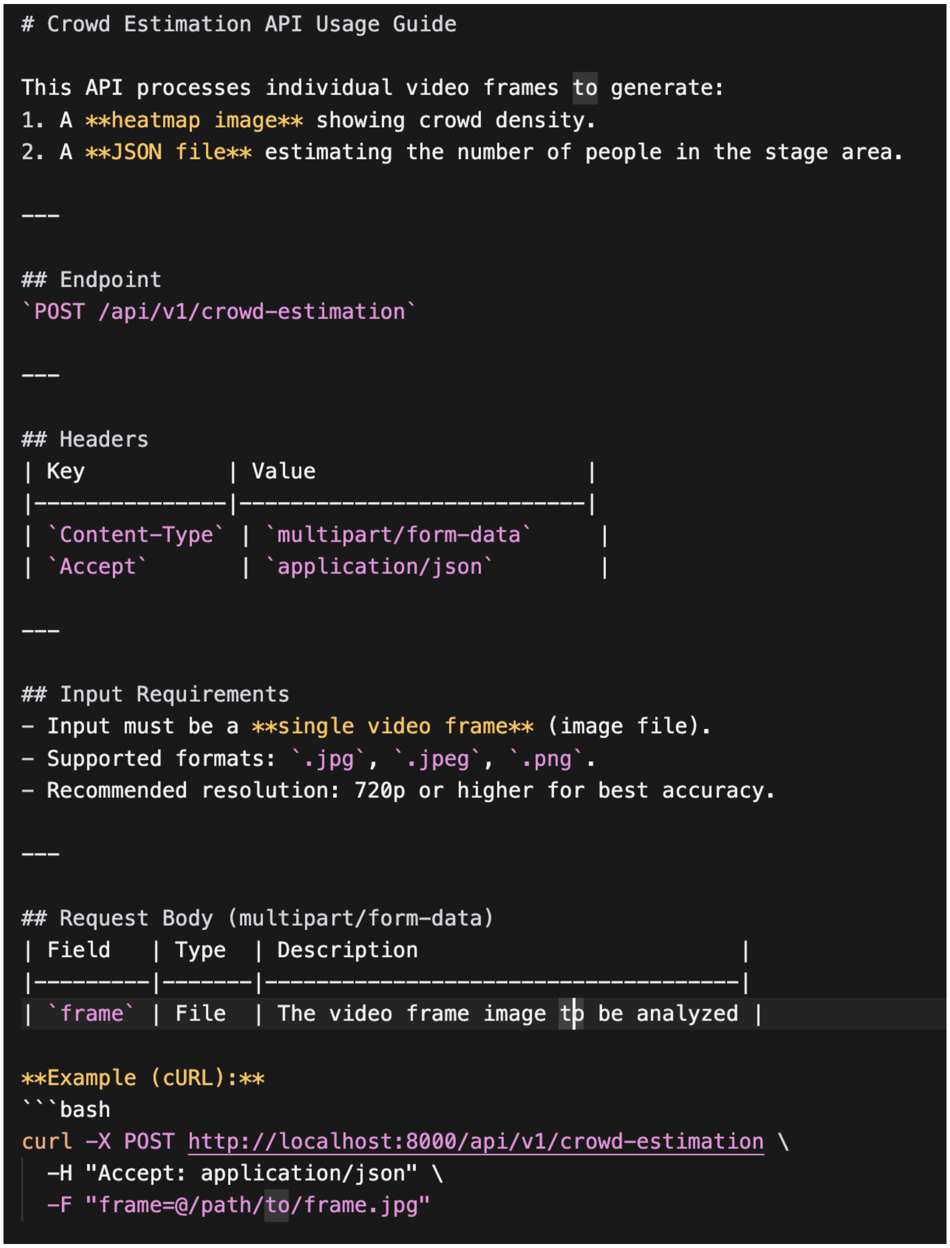
- Finalize and thoroughly test the integrated pipeline and API, ensuring that individual video frames can be processed end-to-end to produce both crowd density heatmaps and JSON-based people counts, and that the system works seamlessly with the backend team for future integration.
- Collect, carefully annotate, and organize frames extracted from AFL footage to create a comprehensive custom dataset for YOLO, ensuring the data is well-prepared for the upcoming training and accurate evaluation of the crowd detection model.

1. **Build a pipeline that integrates both steps and create an API to enable collaboration with the backend team**

Assigned to: Son Tung, Jake

Outcomes:

- Successfully combined **LISA for Audience Segmentation** and **CSRNet for Crowd Density Estimation** into a single pipeline, enabling end-to-end processing of video frames.
- Developed an **API** that takes individual video frames as input and outputs (1) a heatmap image showing crowd density, and (2) a JSON file with the estimated number of people in the stage area.
- Wrote a **Markdown-based API usage guide** documenting input requirements, headers, request body, and expected responses, making it easier for the backend team to integrate and test the system.



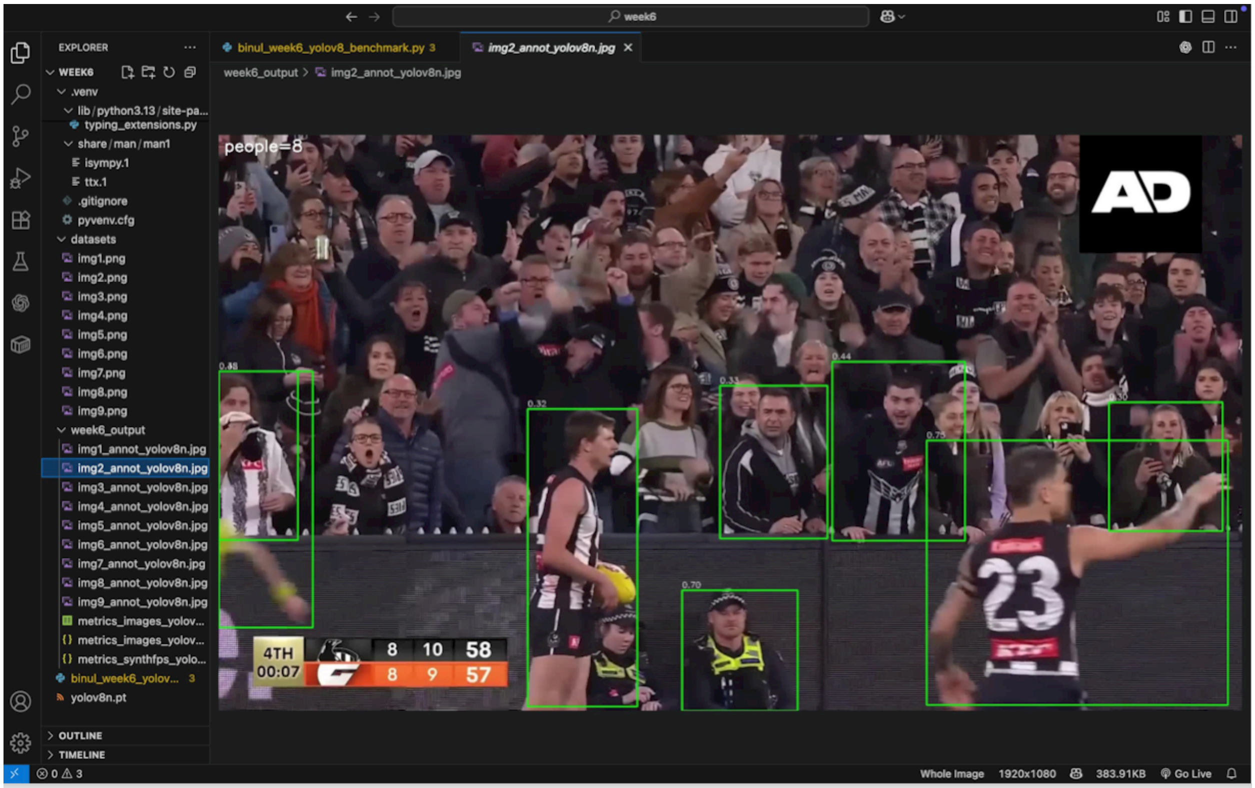
~~~markdown
# Crowd Estimation API Usage Guide

This API processes individual video frames to generate:
1. A **heatmap image** showing crowd density.
2. A **JSON file** estimating the number of people in the stage area.

---

## Endpoint
`POST /api/v1/crowd-estimation`

---

## Headers
| Key            | Value                     |
|---------------|--------------------------|
| `Content-Type` | `multipart/form-data`     |
| `Accept`       | `application/json`        |

---

## Input Requirements
- Input must be a **single video frame** (image file).
- Supported formats: `.jpg`, `.jpeg`, `.png`.
- Recommended resolution: 720p or higher for best accuracy.

---

## Request Body (multipart/form-data)
| Field   | Type  | Description                          |
|---------|-------|--------------------------------------|
| `frame` | File  | The video frame image to be analyzed |

**Example (cURL):**
```bash
curl -X POST http://localhost:8000/api/v1/crowd-estimation \
  -H "Accept: application/json" \
  -F "frame=@/path/to/frame.jpg"
~~~

2. **Train YOLO on frames extracted from AFL footage samples (2 weeks)**

Assigned to: Anika, Saksham, Binul

Outcomes this week:

- Test YOLOv8 for person detection on AFL snapshots with the custom dataset
- Collected and annotated frames extracted from AFL footage samples to prepare a custom training dataset for YOLO.



Note: The results were obtained using models with pretrained COCO weights, not AFL-specific weights.

Plan for next week:

- Trained YOLO on the annotated dataset to detect crowd regions within the frames and evaluated model accuracy using metrics such as mAP.
- Compared model performance before and after training to measure improvements in crowd detection capability.