Object Instance Counting in Videos by MLLMs

By Cari He

# Introduction

In this project, I investigated the capability of closed source Gemini and open-sourced Qwen to count unique object instances on the *Continuous Perception Benchmark* [[1]](https://www.arxiv.org/pdf/2408.07867). Through getting the model to explain its thought processes, I uncovered some common misconceptions the model has, and thereby proposing improvements through in-context learning and prompt tuning to correct these misconceptions. The 2 main discoveries are in model Bias to Grouping and Camera Movement.

# Gemini Experiments

## Bias to Grouping

The original prompt given is as follows:

PROMPT  
“As a proficient video understanding model, your task is to closely observe the objects within the scene in the provided video and determine the total count of {category} present. Please provide your response as a single numerical value, indicating the quantity of {category} observed, without any other word.”

### Identifying Bias in Grouping

By excluding the restriction on the format, I first let the model explain how it got to its answer. Given office1\_cake\_12.mp4 video, which shows 10 tables with varying numbers of cakes and a total of 21 cakes, the Gemini’s response was:

RESPONSE  
“Here's the answer based on the provided video:  
There are a total of 12 cakes in the scene. They're arranged two to a table, and there are six tables.”

Upon assessing multiple videos, I do realize that Gemini tends to conclude its answer by quoting *n* tables and *m* objects per table hence giving *nxm* total objects. This results in the model hallucinating the presence or lack of objects to fit into this framework that all tables have the same number in order to multiply.

### Chain-of-thought

To break away from the concept that all groups (‘tables’) have the same number of objects, I forced the model to give the count of objects on each table before summing them up. I demonstrated the chain of thought with an in-context example.

PROMPT  
"### Task Description  
Count the number of cakes on each table, starting from the first table to the last. After counting the cakes on each table, calculate and provide the total number of cakes across all tables.  
### Example:  
\*\*Task\*\*:  
[sample short video]  
\*\*Table Number\*\* | \*\*Number of Cakes\*\*  
------- | --------  
1 | 3  
2 | 2  
\*\*Total\*\*: 5   
### New Task:  
\*\*Task\*\*:  
\*\*Table Number\*\* | \*\*Number of Cakes\*\*  
------- | --------"

RESPONSE  
Here's the answer based on the provided video:  
\*\*Table Number\*\* | \*\*Number of Cakes\*\*  
------- | --------  
1 | 2  
2 | 2  
3 | 3  
4 | 2  
5 | 3  
6 | 3  
7 | 2  
8 | 3  
\*\*Total\*\*: 20

With the CoT and In-context example, the count improved for office1\_cake\_12.mp4 from the original 12 cakes to now 20 cakes, closer to the ground truth 21 cakes.

### Single Frame Object Instance Counting

A brief consideration was given to assess the model’s ability to count in single frame. The belief is if Gemini can effectively count in a single frame, it therefore can better count in the video. A single frame was given to Gemini to test its ability to count in an image, with poor results. To help it count, I passed the image through a YOLO object detector to draw some grounding boxes to assist the Gemini model in counting. However, due to the perspective of the video, there is indeed much occlusion which results in much overlap between the bounding boxes and more bounding boxes drawn than there are cakes. In this direction of work, maybe using arrows to point instead of bounding boxes could be a good way forward.

A classroom with desks and chairs

Description automatically generated

### In-context Learning

To better understand why Gemini failed earlier, I also prompted Gemini to explain for itself. It gives occlusion, low resolution and lack of resources as its main reasons for failing the task.

The efforts in single frame object counting and Gemini’s reasoning for failure does highlight the key challenge of occlusion. In retrospect, the challenge of occlusion can be better tackled with video instead of an image as it gives various perspectives. Hence to resolve the issue, I adopted in-context learning with a short video snippet and providing count of objects to the video snippet. With time, I may try with YOLO’s bounding box to assist the Gemini model in grounding. It may provide improvements in performance but that means it will require external specialized model which may defeat the original goal of a general-purpose model.

## Camera Movement (Perspective)

The Continuous Perception Benchmark features videos that do a continuous and seamless sweep of the entire room. It is a left to right horizontal movement across the room without angle changes but with varying speeds of movement. This is the first key information the model fails to capture and shown by its response below:

PROMPT  
As a proficient video understanding model, your task is to describe the camera movement in this video.

RESPONSE  
The camera in this video starts in the corner of a room and slowly moves backwards, revealing a classroom with multiple small tables and chairs arranged in front of large windows. The movement is smooth and continuous, providing a steady reveal of the room's contents. The camera stops at the opposite end of the room, then starts moving back toward the initial position.

This response demonstrates why the model may undercount; it thinks that the same object moved out of view and back into view however in actuality the video is continuously moving to the right hence every new object in view was a new object instance.

To overcome this, I simply described the camera movement as it is the same for the entire benchmark. To make this more generalizable in the future, I could use an external module specialized in detecting camera movements.

PERSPECTIVE PROMPT  
In this video, the camera is moving in a straight line from the left side to the right side of the room. The camera is not panning or tilting, it is simply translating along a horizontal path.  
As a proficient video understanding model, your task is to closely observe the objects within the scene in the provided video and determine the total count of {category}s present. Consider whether the {category}s are the same.  
Please provide your response as a single numerical value, indicating the quantity of {category}s observed, without any other word.

By adding perspective, videos such as office1\_book\_12.mp4 where Gemini predicts 12 books whereas there as 24 books are corrected as Gemini misinterpreted the video as reversing back to the start. Gemini is hence able to correctly predict 24 books.

# Qwen Experiments

Qwen shows much poorer ability to count object instances in video, with low number predictions such as 1 or 0, and little explanation.

Qwen unfortunately is less able to follow instructions as Gemini. With the Bias to Grouping in context learning, Qwen defaults to outputting the example’s answer, 5, for all of its responses. The Chain-of-Thought effort did not manage to get Qwen to output in the desired format, hence Qwen did not manage to benefit from the prompt tuning the way Gemini did.

Qwen’s understanding of Camera Movement is very difficult to discern as well. With the same prompt to ask Qwen to explain the camera movement, Qwen’s response is as follows:

RESPONSE  
'The video shows a camera moving around various objects, including a white and blue surfboard with a black and white logo, a white and blue surfboard with a black and white logo, a white and blue surfboard with a black and white logo, and a white and blue surfboard with a black and white logo.

# Gemini vs Qwen Results

As Qwen fails to produce output in a reasonable format with One-shot CoT, I excluded the results for it. The table shows for Gemini, the perspective prompt which adds camera movement information gave improvements in particular to off-by-zero and off-by-one whereas One-shot CoT shows better improvement on Off-by-five metric. This may be due to perspective prompt correcting for specific cases where the model can count effectively but perceived the object instances to be the same. Hence it improves Off-by-zero where with the corrected perception of camera movement, it can count correctly. One-shot CoT on the other hand corrects the way the model derives its answer even when it is less able to count effectively due to occlusion or low resolution. Hence the sample videos it corrects for helps it get closer to the answer but may not be perfect.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Models | Prompt | Off-by-zero (%) | Off-by-one (%) | Off-by-five (%) |
| Gemini | Original | 10 | 24 | 57 |
| Perspective | 13.5 | 28 | 57.5 |
| One-shot CoT | 8 | 25 | 80 |
| Combined | 10 | 27 | 76.5 |
| Qwen | Original | 3 | 7 | 20 |
| Perspective | 0 | 5 | 18 |

# Future Work

My investigations revealed a clear biases and lack of understanding of the model. I think the model fundamentally should have these knowledges imbued either during pretraining or fine-tuning.

For bias to grouping, I think the pre-training dataset should be checked for data that encourages using multiplication to count, and to provide context on which cases one can use multiplication, in other words explicitly indicate the assumption that all tables have the same number of objects to multiply to get the answer. One can provide counting data that does sequential counting with varying number of objects per “group” as well.

For camera movement understanding, we should fine-tune on or train on a new task to help the model to learn to determine camera movement. We could curate data by using camera motion estimators such as using optical flow or key point feature matching to determine camera motion as ground truth. This could help provide context to the models to better understand why objects occasionally disappear and reappear and encourage object permanence upon occlusion.