**PROBLEM STATEMENTS**

1. Predict reaction completion time from video

Problem: Given a video of a chemical reaction (e.g., color change, phase change), predict how much time is left until the reaction is complete.

Why: Useful for monitoring batch processes without physical sampling.

Predicting the remaining time until a chemical reaction completes—based on visual cues like color or phase changes—requires understanding the specific reaction kinetics and conditions. Some reactions, known as "clock reactions," exhibit sudden visual changes after a predictable delay, making them ideal for such predictions.​

**Example:**

**Iodine Clock Reaction**

The iodine clock reaction is a classic demonstration where two clear solutions are mixed, and after a specific time delay, the mixture suddenly turns dark blue. This delay is influenced by factors like concentration, temperature, and the presence of catalysts. By analyzing these parameters, one can predict the timing of the color change.

<https://www.youtube.com/watch?v=R0yn-dbAPHc&ab_channel=TheEfficientEngineer>

**Briggs–Rauscher Reaction:**

An oscillating reaction that cycles through color changes multiple times before reaching completion. The number of cycles and their duration can be influenced by the reaction conditions.

<https://www.youtube.com/watch?v=SCoLMfplVWs&ab_channel=NileRed>

**Belousov–Zhabotinsky Reaction:**

Another oscillating reaction exhibiting rhythmic color changes, useful for studying reaction dynamics.​

<https://www.youtube.com/watch?v=LL3kVtc-4vY&ab_channel=NileRed>

**QUESTIONS**

1. What exactly are we predicting?

Is it a class (e.g., reaction completed: yes/no)?

Is it a continuous value (e.g., time left, viscosity, temperature)?

1. What are the inputs?

Are we using raw video frames?

Or extracted features (like optical flow, frame differences, color histograms)?

How much of the video (full video or short clips)? Clarify data handling.

1. How do we label the data?

How do we know the "ground truth" for each video?

Manual labeling, sensor readings, lab records? (Important to train supervised models.)

1. How big is the dataset?

How many videos do we have?

Are they balanced across classes?

Are videos long or short?

(This will influence model choice — small data = simpler models, big data = deeper networks.)

1. What preprocessing is needed?

Do we need to resize videos? (e.g., all frames 224x224)

Extract frames at intervals?

Enhance video (e.g., denoising, brightness adjustment)?

(This is Essential for clean, uniform input.)

1. What is the timeline of events in the video?

Does the important event (e.g., color change) happen gradually or suddenly?

Is temporal information important?

(If yes → models like LSTM, 3D CNNs, or Transformers that capture time.)

1. Which basic model to start with?

Start simple: Maybe frame-wise CNN classification.

Then add sequence models (e.g., CNN + LSTM) if needed.

1. How will we evaluate the model?

Accuracy? MAE (Mean Absolute Error)? F1-Score?

Do we need early warning (time-sensitive evaluation)?

(Evaluation must match the real-world needs.)

1. Any real-world constraints?

Does it need to work in real-time?

How much delay is acceptable?

Can we tolerate errors (and what kind of errors are most dangerous)?

1. How do we visualize results?

Should we plot prediction vs. true value over time?

Show confusion matrices?

Overlay predictions on videos?