Assignment: Fraud Detection in Retail Transactions Using Computer Vision

Objective: Develop an end-to-end computer vision solution to detect potential fraud during cash transactions in a retail environment, specifically when a cashier fails to generate or provide an invoice. The task includes building an API to simulate invoice generation and integrating the results with MongoDB. Additionally, display counters on the video output showing cash transactions, invoices, and fraud status.

Task Description:

1. **Data Engineering:**
    - You will receive a dataset of static videos showing cash transactions at a retail counter, with some scenarios including the issuance of invoices and others not.
    - Perform data preprocessing, including frame extraction, labeling key actions and data augmentation to enhance model performance.
2. **API Development:**
    - Develop a custom API that simulates invoice generation after a cash transaction. This API should be triggered manually or automatically within your solution and should update a MongoDB database with records of each transaction.
    - Simulate fraud by ensuring that the API is not triggered within 15 seconds of a cash transaction in cases where fraud is to be demonstrated. This delay indicates that an invoice was not generated timely, simulating a fraudulent event.
3. **Model Development:**
    - Create a computer vision model to detect whether an invoice is given after a cash transaction. Use techniques like object detection or action recognition tailored to identify relevant actions in the video.
    - Based on your idea you can use any feature in the given video to train your model to achive this project objective.
4. **Video Output with Counters:**
    - Process the provided videos and overlay a counter on the video output showing:
        - **Cash:** Number of cash transactions detected.
        - **Invoice:** Number of invoices detected following cash transactions.
        - **Fraud:** A flag indicating “Yes” if a transaction is detected as fraudulent (invoice not generated within 15 seconds) and “No” otherwise.
    - The counters should update dynamically as the video plays, providing a clear visual representation of detected actions and fraud status.
5. **Database Integration:**
    - Use MongoDB to store detailed transaction records, including timestamps, cash transactions, invoice status, and fraud flags.
    - Design an efficient schema to support the retrieval and analysis of transaction data, aligning with the business logic and video processing outputs.

Deliverables:

- A GitHub repository containing the complete codebase, including scripts for data processing, model training, API development, business logic, and MongoDB integration.
- Record a video (10-15 minutes) explaining and demonstrating your solution, showing the video output with counters and a walkthrough of the MongoDB integration and API functionality.

Note: Assignment timeline 2 to 3 days.