**Real-Time Motion Capture and Malpractice Detection System**

This project is a real-time motion capture and monitoring system designed for secure online examinations. It captures images during motion detection using IP cameras or webcams, and performs a malpractice detection analysis on the captured images using machine learning techniques. A final report is generated in the form of a bar chart, showing the degree of similarity between captured images and pre-stored malpractice images.

**Features**

* **Real-time motion capture**: Records continuous video and detects motion using IP or USB cameras.
* **Image capture upon motion detection**: Captures images at specific intervals when motion is detected.
* **Malpractice detection analysis**: Compares captured images with a set of training images (malpractice samples) to check for possible malpractice.
* **Report generation**: Saves a bar chart showing malpractice match scores for each image in a report folder.

**Requirements**

* Python 3.x
* Required libraries: OpenCV, NumPy, Matplotlib

You can install these libraries using:

pip install opencv-python numpy matplotlib

**Setup and Configuration**

1. **Clone the Repository**

git clone https://github.com/arunsuriya3702/Real-Time-Motion-Capture.git

1. **Set Up Folders**: Ensure that the following folders are created for the program to work correctly:
   * captured\_images: To save images captured during motion detection.
   * report: To save the generated malpractice detection bar chart.
   * training\_images: Add any known malpractice images to this folder for comparison with captured images.
2. **Edit IP Camera URLs**: The program allows you to input IP camera URLs manually. When prompted, enter the unique URLs for each camera you want to use. Leave the prompt blank if no IP camera is available, and the program will check for USB or built-in webcams.

**Usage**

1. **Run the Program**: Start the program by running:

python your\_script\_name.py

1. **Input IP Camera Addresses**: When prompted, enter the IP addresses for your cameras (e.g., http://192.168.x.x:8080/shot.jpg). If no IP cameras are available, the program will attempt to use USB or built-in webcams.
2. **Motion Detection and Image Capture**: The program continuously captures video. Once the motion detection delay period has elapsed, it begins checking for motion. When motion is detected, it saves images at specified intervals to the captured\_images folder.
3. **Malpractice Analysis**:
   * Press q to stop motion capture and begin the malpractice detection analysis.
   * The program compares captured images with the images in the training\_images folder.
   * A bar chart showing match scores is generated and saved to the report folder as malpractice\_detection\_analysis.png.

**Customization**

* **Adjustable Parameters**:
  + image\_capture\_interval: Set the time interval (in seconds) for image capture during motion detection.
  + motion\_detection\_delay: Set the delay (in seconds) before motion detection starts after the program runs.
  + image\_capture\_limit: Set the maximum number of images to capture.
  + retry\_limit and retry\_delay: Configure the number of retries and delay when attempting to capture a frame.
* **Malpractice Detection Sensitivity**: The program uses ORB (Oriented FAST and Rotated BRIEF) for feature extraction and BFMatcher for feature matching. You can adjust the minimum contour area in the code to filter out small, irrelevant motions.

**Example Output**

The output of the analysis is a bar chart saved as malpractice\_detection\_analysis.png in the report folder, with each bar representing the similarity score of captured images compared to the training images.

**Notes**

* Ensure your camera URLs are accessible and IP addresses are correct.
* The program is designed to help monitor online examinations, but the parameters and sensitivity should be adjusted according to your specific environment.