**Person Detection and Tracking System**

This project focuses on detecting and tracking persons (specifically children with Autism Spectrum Disorder and therapists) in videos. The system assigns unique IDs to detected individuals and tracks them throughout the video, handling re-entries, occlusions, and gaze estimation.

**Table of Contents**

* [Project Overview](#project-overview)
* [Installation](#installation)
* [Usage](#usage)
* [Code Structure](#code-structure)
* [Model Details](#model-details)
* [Optimization Tips](#optimization-tips)
* [Results](#results)
* [Contributing](#contributing)
* [License](#license)

**Project Overview**

The goal of this project is to analyze behaviors, emotions, and engagement levels of children with Autism Spectrum Disorder and therapists by tracking them in video footage. The system uses state-of-the-art deep learning models for person detection and tracking and includes the following key features:

* **Unique ID Assignment:** Assigns a unique ID to each detected person and tracks them throughout the video.
* **Re-entry Tracking:** Accurately tracks a person even after they exit and re-enter the frame.
* **Post-Occlusion Tracking:** Re-tracks individuals after occlusion or partial visibility.
* **Gaze Estimation:** Estimates and visualizes the gaze direction of the tracked individuals.

**Installation**

**Prerequisites**

* Python 3.8+
* A machine with a GPU is recommended for faster processing.

**Setup**

1. Clone the repository:

git clone https://github.com/thepiyush09/Person-Detection-And-Tracking.git

cd Person-Detection-And-Tracking

1. Create a virtual environment and activate it:

python -m venv venv

# On Windows

venv\Scripts\activate

# On macOS/Linux

source venv/bin/activate

1. Install the required dependencies:

pip install -r requirements.txt

**Usage**

1. Place your input video file in the project directory (e.g., input\_video.mp4).
2. Run the main script to process the video:

python main.py

1. The output video with bounding boxes and unique IDs overlaid will be saved as output\_video.mp4 in the project directory.

**Optional Parameters**

You can specify different input and output video files by modifying the main.py script or by passing parameters when running the script:

python main.py --input\_video path\_to\_input.mp4 --output\_video path\_to\_output.mp4

**Code Structure**

* **main.py**: The entry point of the project, orchestrating video processing, detection, tracking, and gaze estimation.
* **detection.py**: Handles person detection using YOLOv5.
* **tracking.py**: Manages tracking using the DeepSORT algorithm.
* **gaze\_estimation.py**: Implements gaze estimation using MediaPipe's Face Mesh.
* **utils.py**: Contains utility functions for bounding box conversion and video frame saving (optional).
* **requirements.txt**: Lists all the dependencies required for the project.

**Model Details**

* **YOLOv5**: Used for detecting persons in the video. YOLOv5 is fast and accurate, making it suitable for real-time applications.
* **DeepSORT**: Used for tracking detected persons across frames, handling re-identification, and occlusions.
* **MediaPipe Face Mesh**: Used for estimating and visualizing the gaze direction of detected faces.

**Optimization Tips**

* **Use a GPU**: If available, use a GPU for significantly faster processing times.
* **Reduce Video Resolution**: Lower the resolution of the input video to speed up processing.
* **Skip Frames**: If real-time processing is not required, consider processing every nth frame to reduce computational load.

**Results**

The system processes the input video and outputs a video with the following:

* Bounding boxes around detected persons.
* Unique IDs for each tracked person.
* Arrows indicating the gaze direction of each person.

These outputs help in analyzing interactions and engagement in the context of therapy for children with Autism Spectrum Disorder.