# Advance Computer Vision

CSCI.731.01

CS731_HW_NN_2225_Background_Subtraction

**Aashka Desai**

1.

**Collaboration :** For this Assignment, I have taken some help from the C++ documentation as well as help of the generative AI Chatgpt to understand the implementation of the tasks.

I have collaborated with a few fellow batchmates to discuss the implementation and understand their approach in applying the background subtraction to the videos. It was very interesting as well as insightful to know how they used their knowledge and understanding to implement this assignment.

During our discussion, Abhinav shared their approach to applying the MOG2 background subtraction method, which was very insightful. We discussed a few problems that arose while creating the background image as well as while copying the mask to the original video during background subtraction.

I also got great insights from Aditi Karad on troubleshooting the video saving issues as well as playing around with different parameters in an attempt to refine the method and results obtained for the background replacement.

2.

**Create a model of the background of the given video**


Fig. Background Image for VIDEO_Walking_through_Back_Yard_DO_NOT_REMOVE.mp4

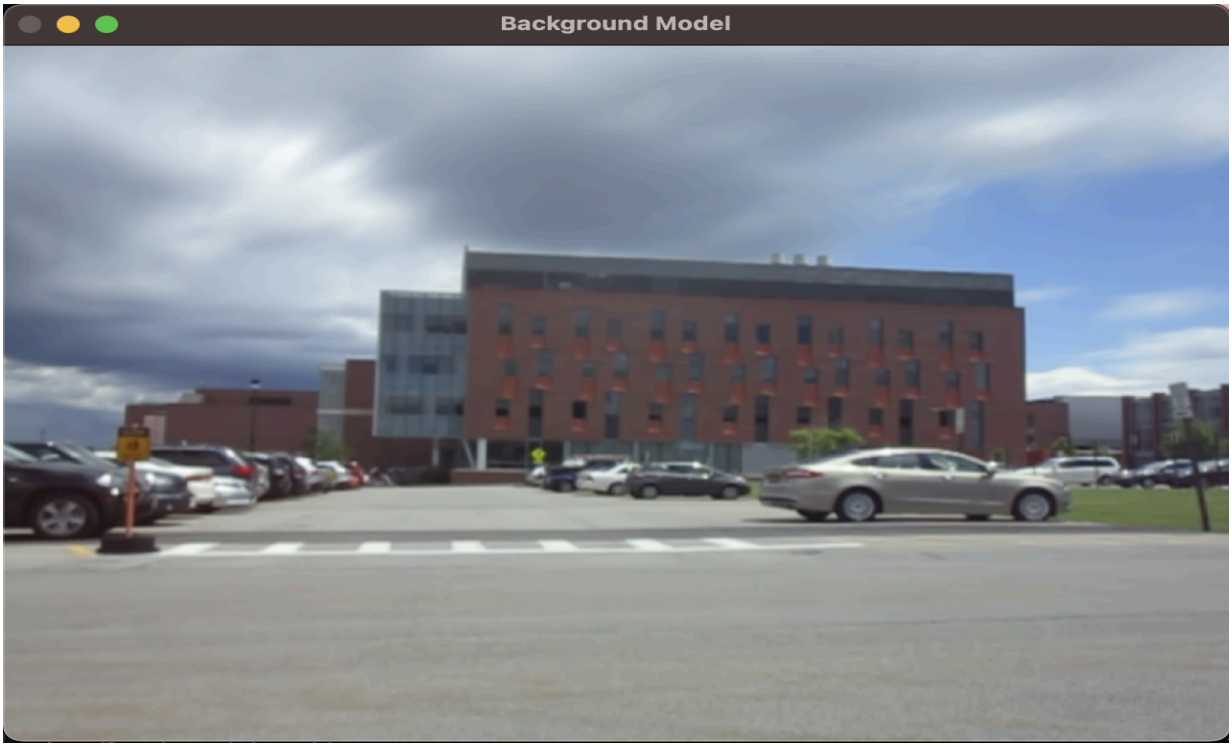
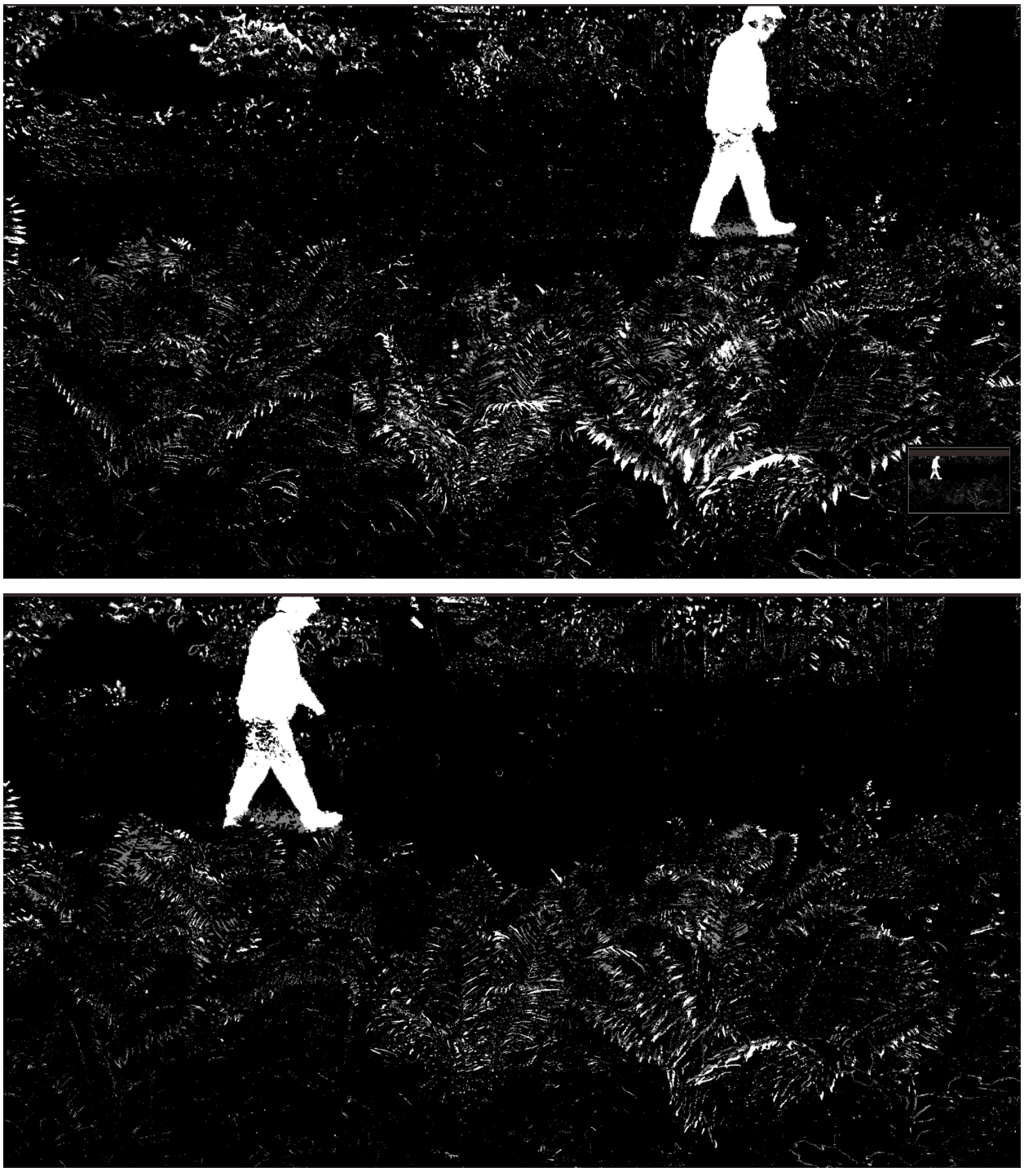
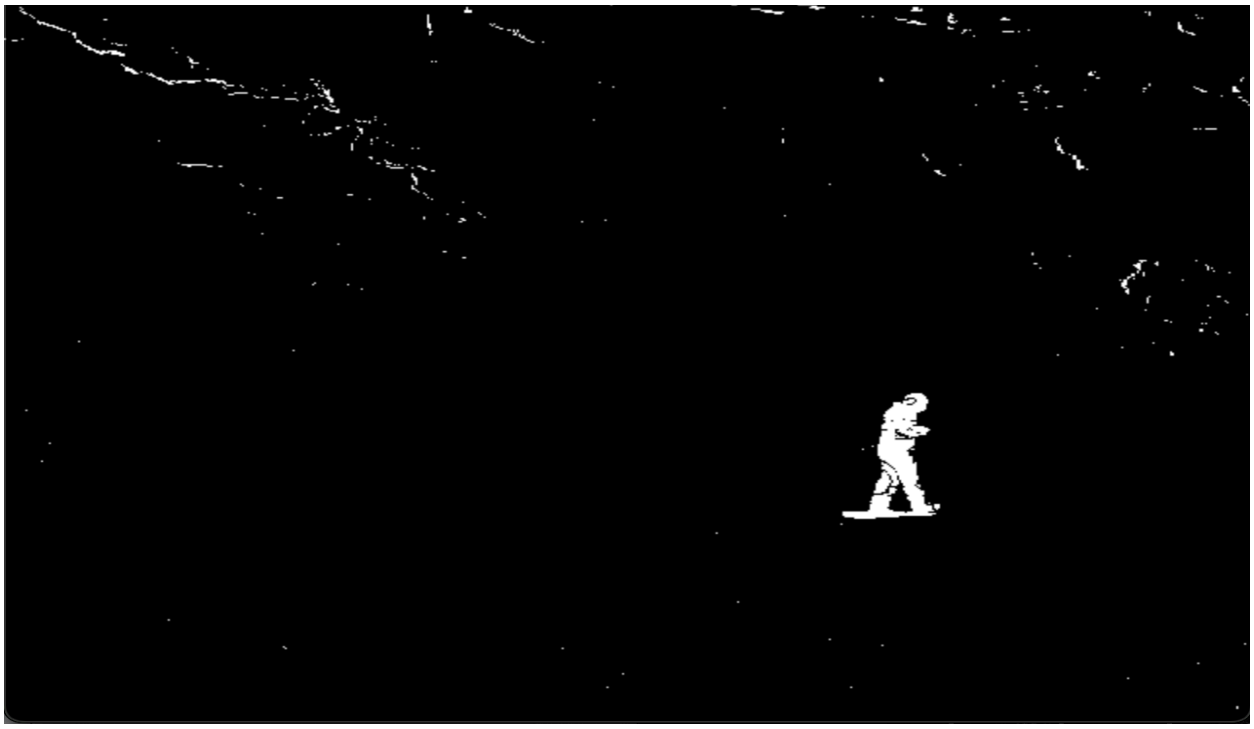


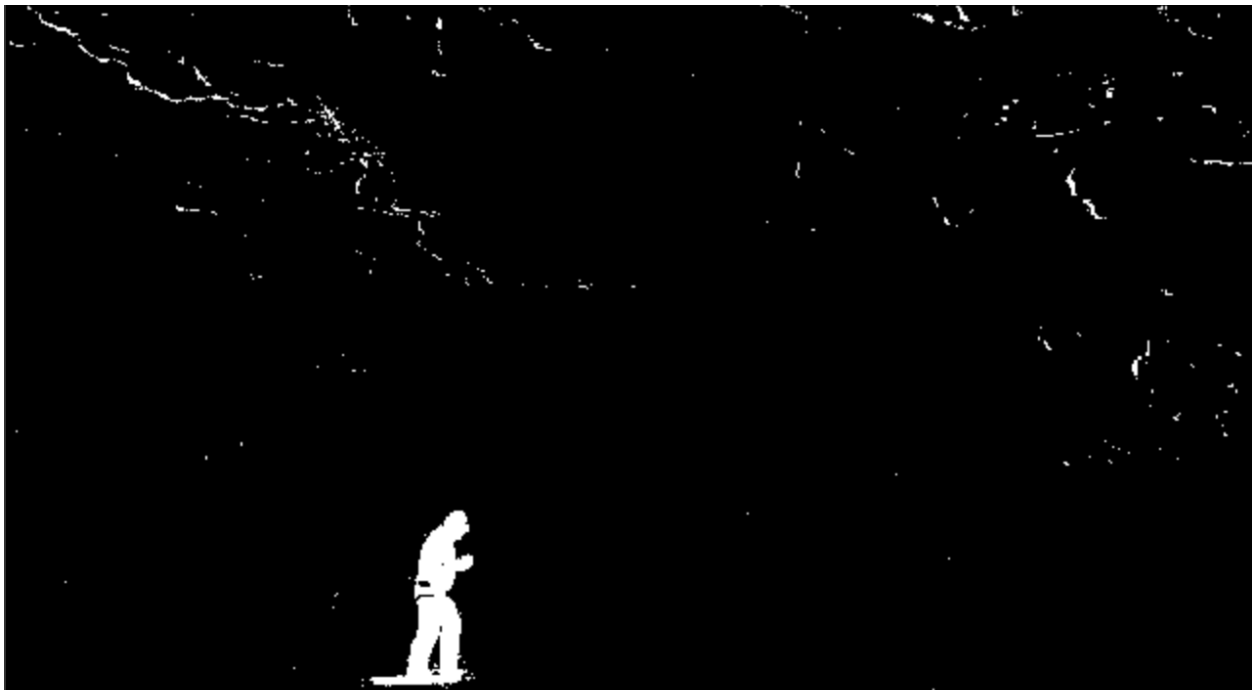

Fig. Background Image for DR_PEDESTRIAN_2017_06_20_MOV1.MOV

Using createBackgroundSubtractorMOG2 on
VIDEO_Walking_through_Back_Yard_DO_NOT_REMOVE.mp4 to get Foreground Mask

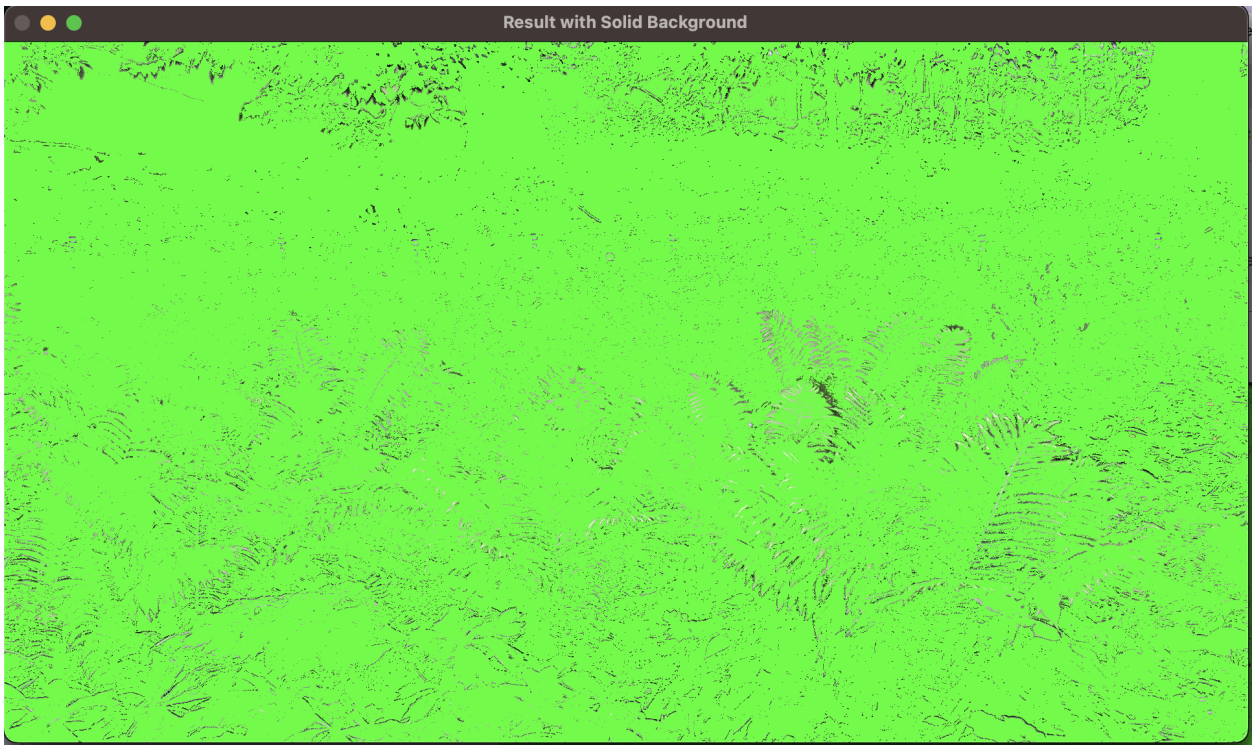


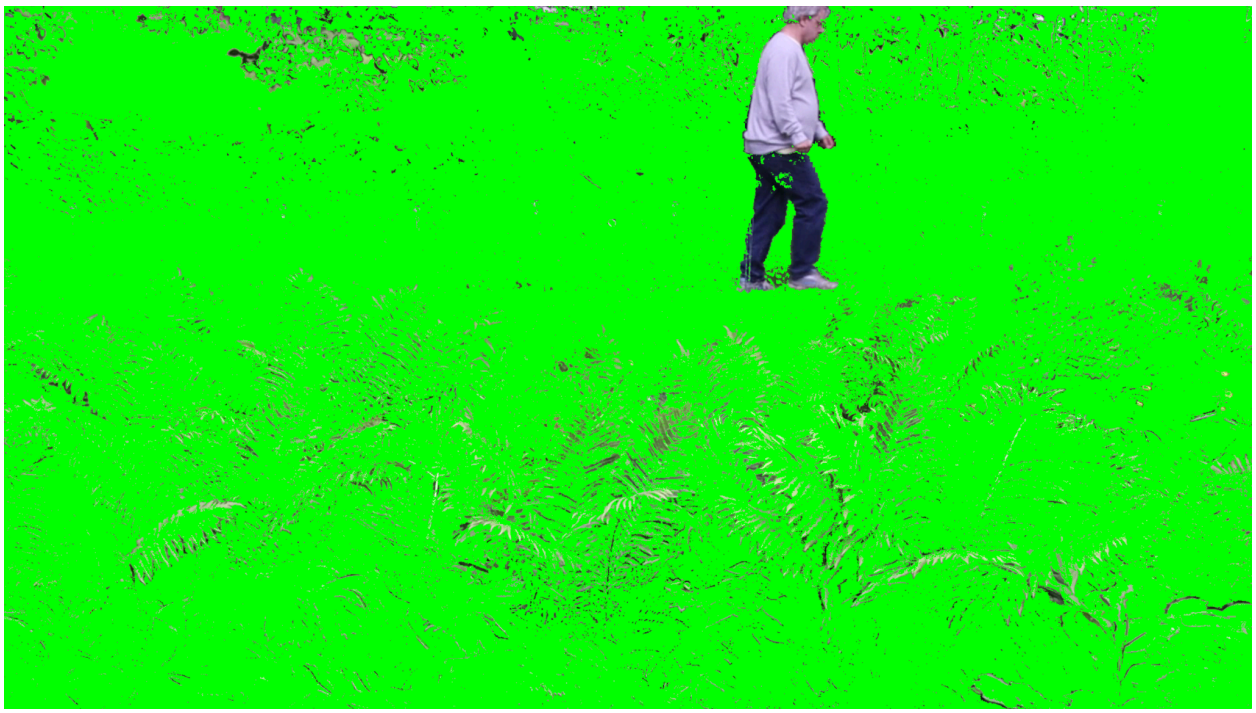
Using createBackgroundSubtractorMOG2 on
DR_PEDESTRIAN_2017_06_20_MOV1.MOV to get Foreground Mask



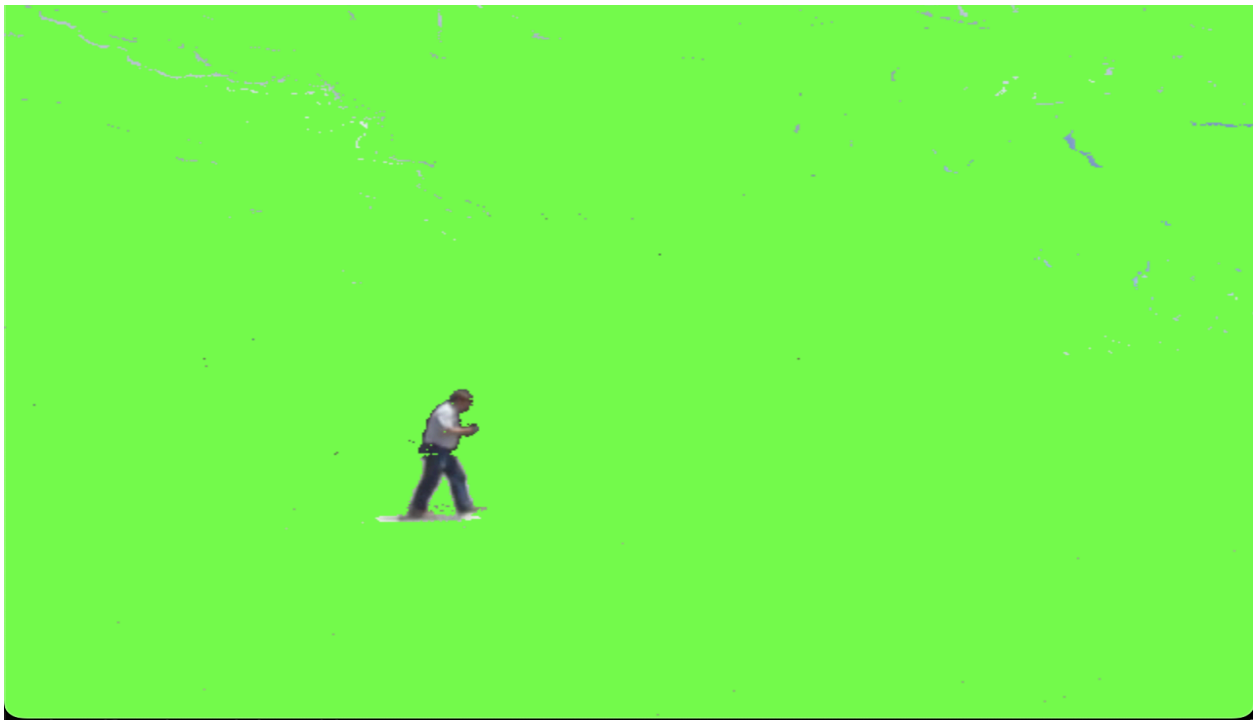


3.




Result with Solid Background



4.

To create a background model of the video provided, I utilized the `BackgroundSubtractorMOG2` from OpenCV. The video's initial 10 seconds of inactivity were critical, as this period allowed the algorithm to learn the background without any moving elements, like Dr. Kinsman, thereby establishing a stable model. This model serves as a reference for the algorithm to detect changes in subsequent frames.

As I processed the video frame by frame, `pMOG2->apply()` was called to update the foreground mask, which differentiates between the static background and moving objects. The `pMOG2->getBackgroundImage()` function was then used to retrieve the background model for the current frame, which I displayed using `cv::imshow()`.

To isolate Dr. Kinsman from the background, I created a new `cv::Mat` object filled with a solid color using the `solidBackgroundColor` variable. The color choice can be altered by changing the `cv::Scalar` values. For each frame, the foreground mask obtained from the MOG2 subtractor was used to copy Dr. Kinsman's image from the original frame onto the solid colored background, effectively removing the original background.

The application of this method resulted in a series of frames where Dr. Kinsman is clearly visible against the newly colored backdrop. This demonstrates the MOG2 algorithm's effectiveness in segmenting moving objects from their background. For reporting purposes, I chose key frames that best illustrated the successful application of this technique. These frames were saved and displayed in the write-up to provide visual evidence of the isolated subject against the solid background color.

Throughout the process, real-time display windows were used to show the original frame, the foreground mask, the background model, and the final result with the solid background, which was crucial for fine-tuning the algorithm's parameters and visually verifying the output.

In conclusion, this homework has been an enriching learning experience in the field of computer vision and specifically in the application of background subtraction methods. Working with the `BackgroundSubtractorMOG2` class in OpenCV, I've gained a deeper understanding of how background models are constructed and the nuances involved in differentiating between foreground and background elements. The process of tweaking the algorithm parameters to accommodate various video characteristics has taught me the importance of customization in computer vision tasks.

One of the key lessons learned is the significance of the initial learning phase for the background subtractor. Those first 10 seconds of static background were instrumental in creating a clean and stable background model. This highlighted the need for a careful setup of the scene and recording conditions when capturing video for analysis. I also encountered the computational demands of video processing, which prompted me to explore more efficient ways to process and display video frames. This is a note to my future self to always consider the

performance implications of the chosen algorithms and the potential need for hardware acceleration or optimized code.

Lastly, the value of collaboration cannot be overstated. Engaging with my classmates provided different perspectives and solutions to common problems, such as dealing with shadows and dynamic backgrounds. It was a reminder that while individual effort is crucial, the collective knowledge and experience of a group can propel you further. I learned that discussing and validating my approach with others leads to a more robust and well-rounded solution. For future projects, I aim to incorporate this collaborative spirit from the outset and maintain a balance between self-reliance and teamwork.