# 3. DESIGN

Understanding Main Logic

When we talk about a video, it means that we are talking about frames or a stack of pictures, and together these moving frames create a video. This program is basically designed by us to capture different frames (stack of pictures) and compare them with the initial static frame of the video where no movement is detected. That's how we can detect the difference in positioning and movements of objects between the initial static frame where no movement is detected and random capturing of frames through using different filters in the webcam of our system. This method is not only helpful for us to detect the change in the position of objects present in the initial frames but also helps us to track the movement through the proctoring process.. The two images or frames we compare after capturing them from the webcam of our system are actually compared based on the pixel intensity of the images, and this pixel intensity difference helps us find out the difference between the two images. For this, we will design a Python program, as we have discussed already in this tutorial, in which we will use functions of pandas, cv2, and many in-built modules so that we can compare the pixel intensity of images captured through the webcam of our system. After comparing the pixel intensity and finding out the intensity differences in images captured, we can detect the movement of objects in the frame. This is the main logic of the program we will write where we use the functions so that we can capture the differences and compare the pixel intensity differences using the program itself.

The Program

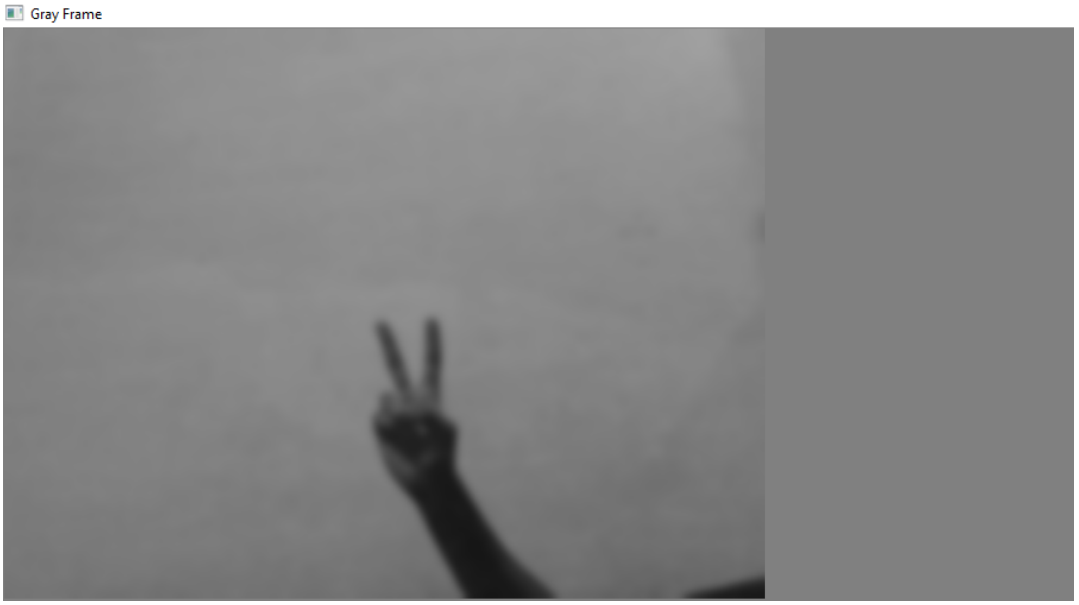
**Output:**

When we run the program, four different types of video frames will be opened on the output screen, which will capture the movements of an object present in the initial frame and added to the video frame later on. And apart from the frames, a CSV file will all the motion timing details will be saved in the same directory where the code is present, but we will discuss this only after understanding all the four output video frames. These frames have their own specialty, and we will describe all of them by explaining the functioning of each of these frames. Each of these frames will be used to capture the movements and work like a motion detector using our system's webcam. These 4 video frame windows will capture different types of results, and each of them has different output, and therefore, we will study each of them separately as the output of the program. **Following is the description of the different color frames which will be displayed to us in the output window when we execute the program:**

**(1) Gray Frame:** The first video frame that will open in the output screen is the gray frame. The image displayed inside the gray frame is displayed in the grayscale or gray shade (That's why this frame is named the gray frame), and it is also a bit blur. As we know, RGB (Red, Green & Blue) image frames have three intensity values. But, there is only one intensity

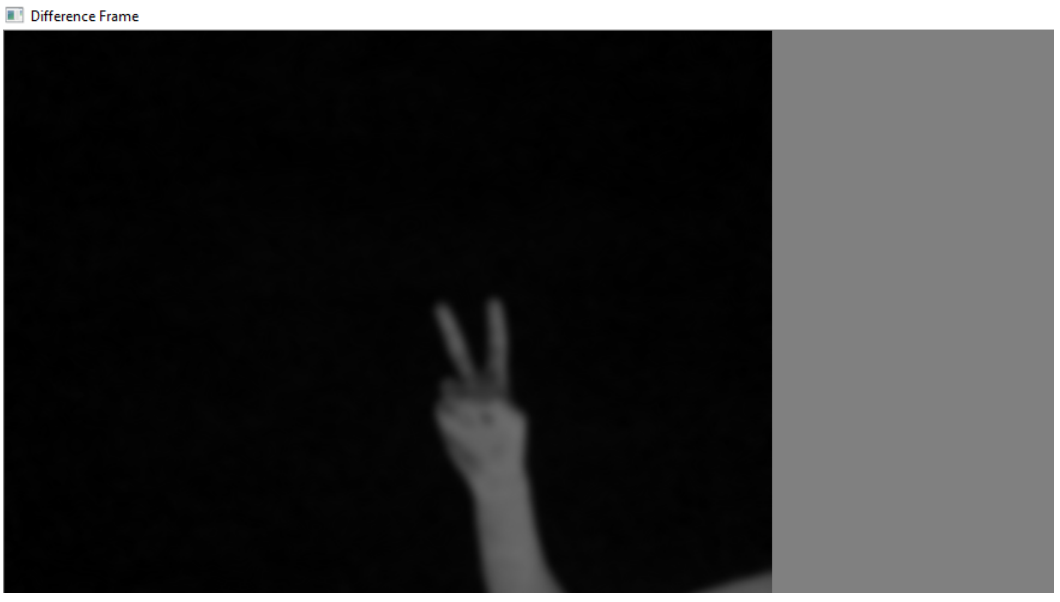
value in the gray picture frame, as displayed here. This makes it even easier to capture the intensity value differences for the object present in the gray frame.

**A resultant output window for the gray picture frame is displayed as the output of the program:**



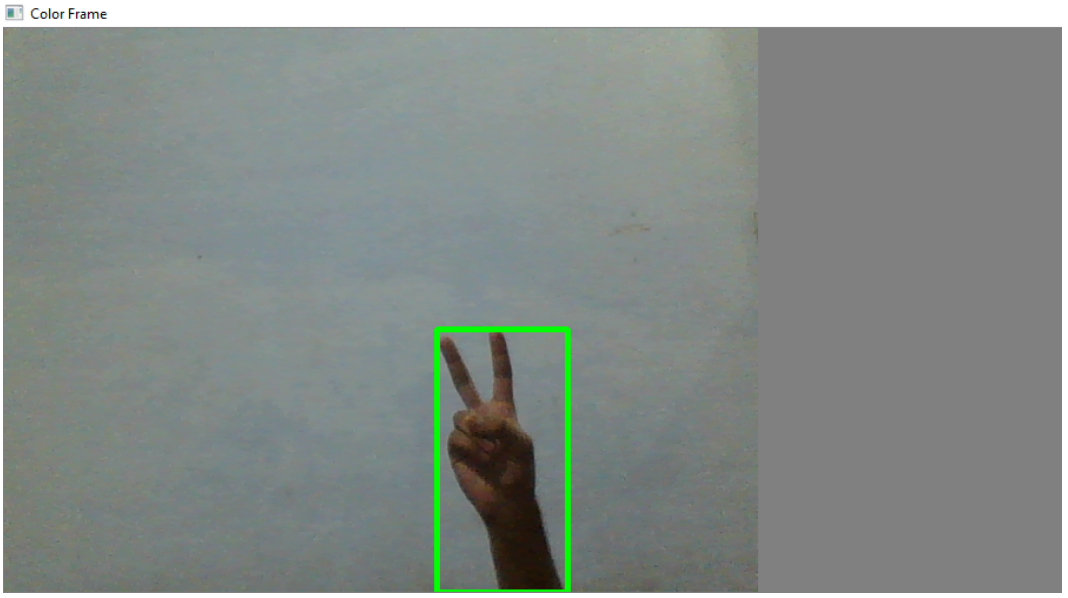

**(2) Difference Frame:** Instead of naming this frame as a black & white frame, we have named this frame as a difference frame because we will capture the difference of intensity values in the initial static frame & current frame through this video frame. Through this frame, we can detect the motion in the frame as well as movements of objects in the frame using the difference of intensity values in the initial static frame and current frame.

**Following is the resultant difference frame displayed to us in the output screen:**



**(3) Colour frame:** This is the colour frame where images are displayed in the colour shade format (as we usually see in a camera or system's webcam). We capture the colour images in the color frame, and the movements of the objects present in the colour frame are highlighted with the green rectangle around the area where movement is happening in the frame.

**A resultant output window for the color picture frame is displayed as the output of the program:**




**(4) Threshold Frame:** This is the GaussianBlur picture filter frame, the filter which we applied in the program after the gray picture filter. As we defined in the program, when the intensity of the pixel changes with a difference of more than 30, then the pixel will be highlighted with the white color. In the opposite case, when the pixel's intensity changes with a difference of less than 30, then the pixel difference will be highlighted with the black color.

**Following is the resultant Threshold frame displayed to us in the output screen:**

Threshold Frame