Image Convolution

- **The used image:**

image

- **Mean Filter:**

The Mean Filter or Average Filter is a linear smoothing filter where it takes each pixel and looks at its neighborhood (e.g., 3×3 region) then replaces the pixel’s intensity with the average of all those values, so for each pixel we replace it by the mean of itself and its 8 neighbors.

We did first convert the RGB image to grayscale using this formula

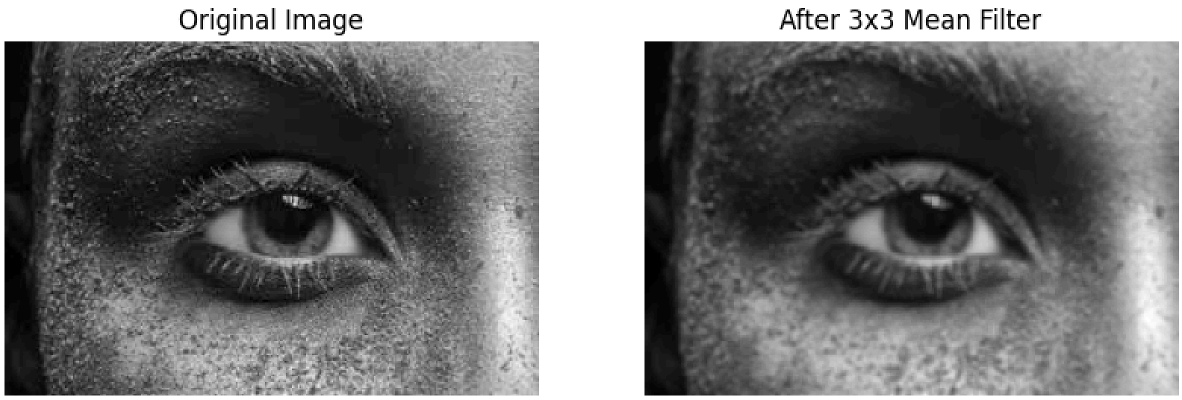
$$Gray=0.2989R+0.5870G+0.1140B$$

to get a 2D matrix which represents the luminance of each pixel

After creating the kernel, each element will be 1/9 because there are 9 pixels in a 3×3 area, and when applied, it will sum all the 9 pixel intensities and divide by 9 to get the average.

Now, at each position, the kernel is placed over a small neighborhood of pixels as a 3×3 region around the current pixel. Each value in the kernel is multiplied by the corresponding pixel intensity beneath it, and all these products are then summed together to produce a single number. This resulting value replaces the intensity of the central pixel in the output image. The kernel then moves step by step over the entire image, repeating this process for every pixel location.

Before the filter, the image has sharp edges and clear details but after applying the mean filter, all edges become less sharp and the overall image looks blurred, this happens because averaging spreads intensity values and removes rapid changes (high frequencies) in the image.



- **G2 and G3 kernels:**

G2 Kernel: A filter that multiplies the image intensities by 2 is simply a scalar kernel where only the central pixel is considered and its value is doubled with no neighborhood averaging.

G3 Kernel: Sharpening works by adding details (edges) back to the image that were lost due to blurring so we can define G3 in terms of G1 and G2 as: $G3=G2-G1$

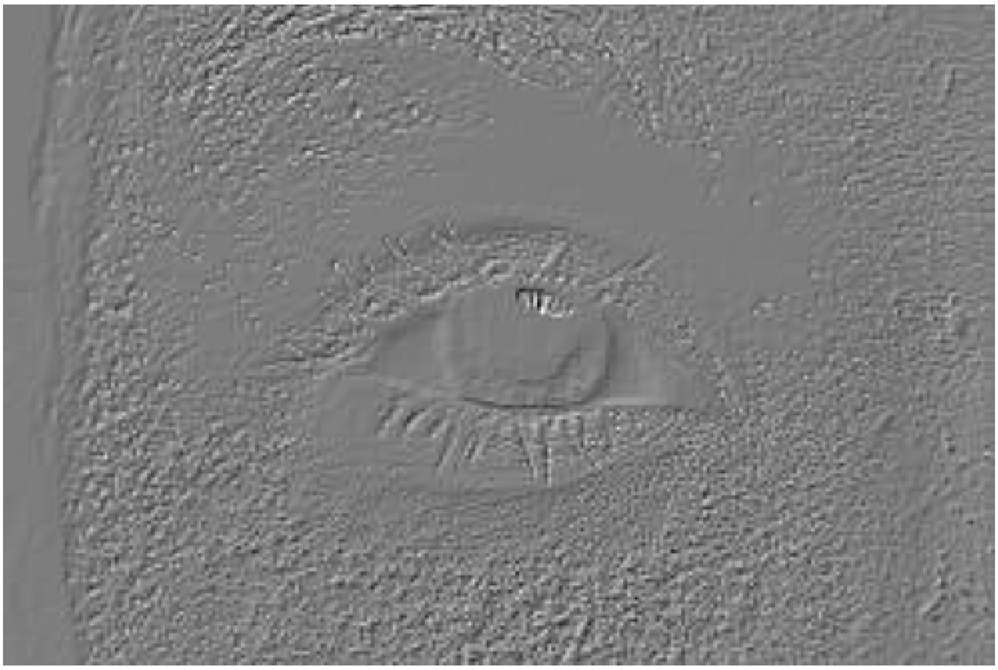
The G2 filter increases overall brightness, while the G3 filter enhances edges and fine details, making the image appear crisper yet slightly brighter, so the convolution filters can manipulate image features from uniform intensity scaling (G2) to controlled edge enhancement (G3).

- **G4 Kernel:**

The G4 Kernel is an edge-enhancement (high-pass) filter, the weights include negative values and the sum of all elements equals 0, so the filter doesn't preserve the average brightness of the image.

When you use *cv2.imshow()* before normalization, the result appears mostly black because the filter's coefficients sum to zero which means positive and negative contributions from pixels cancel out and the output values are centered around zero, not the standard [0, 1] or [0, 255] range that imshow() expects.

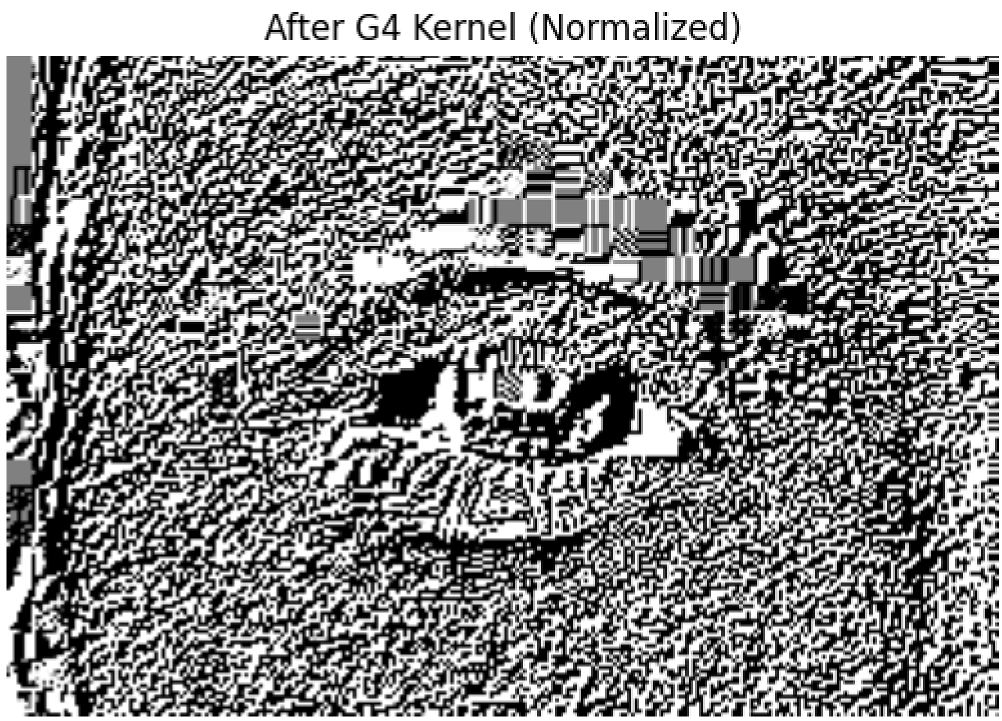
After G4 Kernel



Adding 0.5 recenters the pixel values from around zero to around mid-gray as follows:

new value = old value + 0.5

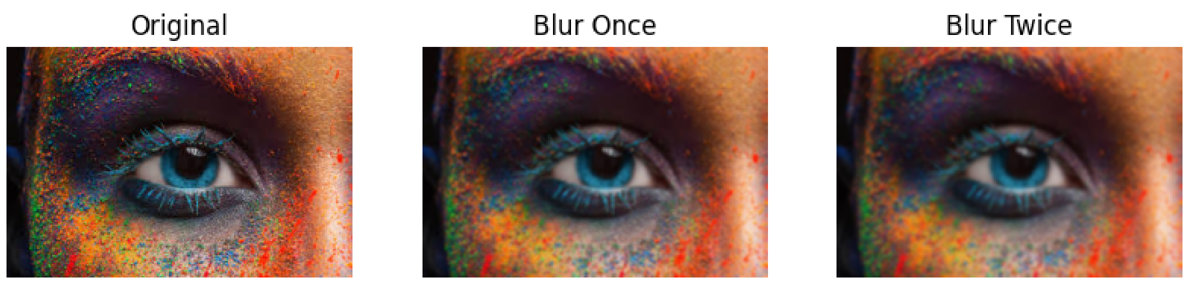
That shifts negative values into a visible range (0 for black and 1 for white), so instead of the entire image being near zero (black), we see both bright and dark regions corresponding to edges or areas of intensity change.

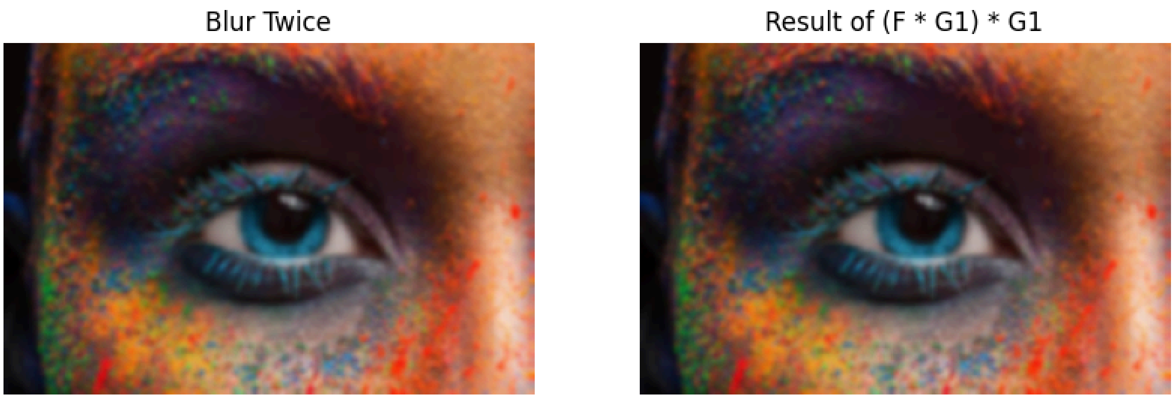


- **Blur Twice:**

The twice blurred image is smoother, edges are softer and less defined. The operation (F * G1) * G1 is equivalent to convolving the original image F with a larger blur kernel (mathematically, it's like using G1 * G1 as a single kernel) so blurring twice or using G_combined which combines G1 and G1 at once will give the same result.

In Gaussian filters, this is equivalent to increasing the standard deviation σ, producing a more intense blur.





- **Sobel Filter:**

The Sobel filter highlights intensity changes in specific directions where the horizontal edges detect changes along rows (left-right) and the vertical edges detect changes along columns (top-bottom), these filters compute image gradients, areas with high intensity change become bright, flat regions become dark.