**Documentation for Machine Problem 2**

In this assignment, I delved into fundamental image processing techniques using OpenCV and Python, specifically focusing on scaling, rotating, blurring, and edge detection.

I began by installing and importing the necessary libraries: `cv2` for OpenCV, `numpy` for numerical operations, and `matplotlib` for displaying images. I used Google Colab’s file upload feature to load an image, which I converted from RGB to BGR color space using `cv2.COLOR\_RGB2BGR`.

For the scaling and rotating tasks, I implemented functions to resize the image by a factor of 0.5 and to rotate it by 45 degrees. The scaling function used `cv2.resize` with linear interpolation to maintain the aspect ratio, while the rotation function used `cv2.getRotationMatrix2D` and `cv2.warpAffine` to rotate the image around its center.

In exploring different blurring techniques, I discovered significant differences between Gaussian blur, median blur, and bilateral filter. The Gaussian blur, applied with an extreme kernel size of (111,111), tends to maintain some level of image identity even when heavily blurred. This is because Gaussian blur smooths the image using a weighted average, which can still preserve overall shapes and structures to some extent.

Conversely, median blur, when applied with an extreme kernel size, significantly obscures the image. This technique replaces each pixel with the median value of its neighborhood, which effectively blurs the image to the center and makes it difficult to recognize distinct features under heavy blurring.

The bilateral filter, on the other hand, proved to be the most effective in retaining image details even with extreme blurring. This technique smooths the image while preserving edges, which allows it to blur the image slightly without losing the important structural information. This balance between noise reduction and edge preservation makes the bilateral filter particularly useful for maintaining the clarity of the image’s details.

Finally, I applied edge detection using the Canny algorithm, which highlighted the edges by detecting rapid changes in intensity. This helped in outlining the contours and structures within the image.

Throughout the assignment, I displayed images at each stage using custom functions to visualize the effects of the transformations. I documented each step in detail, explaining the techniques used, the differences observed, and the results achieved. This hands-on experience provided valuable insights into how different blurring techniques impact image clarity and detail preservation.