**EDGE DETECTION WITH MATHLAB**

**Image with Signature and Edge with Robert Function**

The 'imread' function in MATLAB was utilized to load both the image and the signature. Figure 1 illustrates several operations: the overlaying of the signature onto the image, the conversion of the image to the HSV (Hue, Saturation, Value) color space, and the application of Robert's edge detection method to generate the edges of the HSV image.

**Edge Operators**

In Figure 2, the image was initially transformed to a grayscale. Following that, edge detection was carried out using various operators, specifically Robert, Canny, Laplacian of Gaussian (LoG), Sobel, Prewitt, and Zero-cross.

**Color Segmentation**

Figure 3 demonstrates the process of color segmentation, which was conducted after establishing a threshold value of 0.7. Subsequently, red segmentation was utilized for edge detection.

**Threshold Masked Face**

The image was processed using the 'improfile' function to establish edge identification based on a specific RGB color threshold. The conditions meeting these criteria were then captured in 'im\_edge'. Subsequently, 'im\_edge' was transformed into grayscale to create an edge-detected gotten from the profiled image. Next, the original image was converted to the HSV color space, and threshold conditions were applied to each layer of HSV to generate a binary mask, assigned to 'Lface'. This mask assists in pinpointing the facial features within the image. In order to display these features, 'Lface' was multiplied by 255, effectively isolating the face within the image and storing it as 'imFace'.

Further refinement of the image involved the profiling process to capture only the facial portion by utilizing the 970th row of the image. These steps resulted in the creation of Figure 5, which shows the processed image outcomes.

**Kmeans on RGB and HSV, with Scharr operator for Edge Detection**

The image in the HSV color space was transformed into an unsigned 8-bit integer format necessary for the execution of the k-means clustering algorithm. The algorithm subsequently segments the image into various regions with distinct numbers of clusters (k = 2, 3, 4, and 5). The outcome of each segmentation operation was integrated with the original image, generating visual depictions of the segmented image (labelled as B2, B3, B4, and B5). Following the segmentation process, the third class (k = 3) from the k-means segmented image is selected for edge detection. This process was executed using the Scharr operator.

**SVM on HSV Image**

The image was reduced in size and transformed into HSV color space. It was then reshaped into a 2D matrix where each row represented the Hue, Saturation, and Value components of each pixel in the image. To match the expected input for the SVM model, this matrix was transposed. A binary label was assigned to each pixel, based on its hue value. Using these binary labels and HSV values, an SVM model was trained. This model employed a Radial Basis Function (RBF) kernel, configured to standardize the image and automatically scale the kernel function. Based on the trained SVM model, pixel labels were predicted. If a pixel's score was less than zero, its HSV values were set to zero. These predicted HSV values were then reshaped to the original image dimensions and converted back to RGB color space.

In summary, edge detection was successfully and accurately accomplished by utilizing the Roberts function, color segmentation, and threshold segmentation to define the image's edges with precision and clarity. The detailed profiling of the image likely contributed significantly to the accuracy of this edge detection, as it allowed for the extraction of critical details from the image.