code is captured by a camera device, its luminance value is needed to derive the embedded text. This work thresholds the background image on its luminance. The RGB to YUV conversion is easy and derived using following equation:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100B \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1}$$

3.2 Binarization

Binary images are derived by using threshold on luminance values of images. This should be noted that there is no standard binarization method defined particularly for QR code. This work first divides the image into non-overlapping blocks $b_{i,j}$ of $m_{size} \times m_{size}$ where a block is referred as module and m_{size} does the side of the module. m_{size} is defined as 9 in this work. Then, the average luminance of each block is calculated as follows:

$$\hat{y}_{i,j} = \frac{1}{m_{size} \times m_{size}} \sum_{i=0}^{m_{size}} \sum_{j=0}^{m_{size}} Y_{i,j} \quad \text{for} \quad (i,j) \in b_{i,j} \tag{2}$$

After computation of average luminance of each block, the threshold value for binarization is calculated based on $\hat{y}$ values for window w of blocks. The window size, w_{size} is considered in this work as 5 meaning that each window consists of 5×5 blocks. The procedure for calculation of threshold is given below:

$$Th = \frac{1}{w_{size} \times w_{size}} \sum_{p=i-2}^{i+2} \sum_{q=j-2}^{j+2} \hat{y}_{i,j} \quad \text{for} \quad (p,q) \in w_{i,j} \tag{3}$$

Combining equation (2) and (3) with values of m_{size} and w_{size}, the general expression for computation of threshold for binarization of a pixel in an image can be formulated as follows:

$$Th = \frac{1}{25 \times 81} \sum_{p=i-2}^{i+2} \sum_{q=j-2}^{j+2} \sum_{i=0}^{9} \sum_{j=0}^{9} Y_{i,j} \quad \text{for} \quad (p,q) \in w_{i,j} \tag{4}$$

Finally, the binary image $I_B[i,j]$ is determined by comparing the luminance value of each pixel with the derived threshold value, $\bar{Y}$ as follows:

$$I_B[i,j] = \begin{cases} 1, & \text{if } Y_{i,j} > Th, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

3.3 Weight Calculation

This section introduces weight function $W_{i,j}$ for finding the perceptually important region of background image. Recent studies showed that noisy regions of an image are less sensitive than even regions when pixels of that image are modified. In this work, it is assumed that the center region of an image is more important than that of its surroundings. The weight function is formulated considering these characteristics of images. This work uses variance of pixels to obtain the weight matrix $W_{i,j}$. As variance shows how a pixel varies from its neighboring pixels, variance can be used to classify noisy and noiseless regions of the image. The weight matrix $W_{i,j}$ is calculated as follows:

$$W_{i,j} = \frac{1}{m_{size} \times m_{size}} \sum_{i=0}^{m_{size}} \sum_{j=0}^{m_{size}} \{Y_{i,j} - \hat{y}\}^2 \tag{6}$$

where $\hat{y}$ is calculated as

$$\hat{y} = \frac{1}{m_{size} \times m_{size}} \sum_{i=0}^{m_{size}} \sum_{j=0}^{m_{size}} Y_{i,j}$$

It is worth to be noted that the indexed position in the matrix $W[i,j]$ corresponds to that of QR code.

3.4 Image Blending

In this phase of implementation, the binary image I_B is blended with standard QR code. First, the standard B/W QR code Q_s is generated from given text message. In this work, an open source QR code java library by Zxing [20] is used for the purpose. Then Q_s is blended with the binary image I_B achieved in section 3.2 to generate the binarized QR code Q_b. The value of each pixel in Q_b is derived from I_B and Q_s as follows:

$$Q_b[i,j] = \begin{cases} Q_s[i,j], & \text{if } Q_s[i,j] \in P_{inf}, \\ I_B[i,j] & \text{Otherwise.} \end{cases} \tag{7}$$

where P_{inf} represents the set of version information, format information, finding pattern, alignment pattern, timing pattern and data codewords for input. Q_b, Q_s and I_B presents binarized QR code, standard code and binarized image respectively.

3.5 Color Adjustment

In this section, first, the visual appearance of QR code is improved through the modification of luminance value of image. Then, color QR code Q_c is obtained by converting image I_{YUV} from YUV color space to RGB color space. Binarized Q_b comprises the pixels from binarized image $I_B[i,j]$. Pixels presenting function patterns and other format information remain unchanged. Let $\beta_{i,j}$ be the pixels presenting RS blocks put on the matrix of QR code and satisfies the B/W modules in the respective positions. The modification of luminance value of Q_b is represented as a fuction of the luminance of original image $Y_{i,j}$, luminance threshold derived in (3), $\beta_{i,j}$ and gaussian weight $G_w[i,j]$ where $G_w[i,j]$ is defined as follows:

$$G_w[i,j] = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma}} \times Th \tag{8}$$

where $\sigma = \frac{m_{size}-1}{6}$ in this work. The luminance of pixels $Y_{i,j}$ are modified by the following rules:

$$\begin{aligned} &\text{if } \beta_{i,j} = 1 \\ &Y_{i,j} = \begin{cases} Y_{i,j}, & \text{if } Y_{i,j} > Th + G_w[i,j] \\ Th + G_w[i,j] & \text{otherwise.} \end{cases} \\ &\text{if } \beta_{i,j} = 0 \\ &Y_{i,j} = \begin{cases} Y_{i,j}, & \text{if } Y_{i,j} < Th - G_w[i,j] \\ Th - G_w[i,j] & \text{otherwise.} \end{cases} \end{aligned} \tag{9}$$

At the final phase of implementation, the modified luminance component $Y_{i,j}$ along with untouched U and V components are translated back to RGB color. The YUV to RGB conversion is implemented as follows:

$$\begin{aligned} R &= Y + 1.13983 \times V \\ G &= Y - 0.39465 \times U - 0.58060 \times V \\ B &= Y + 2.03211 \times U \end{aligned} \tag{10}$$