**Image Processing**

Part 1: Image Processing methods-

We created a (1) RGBImage class and a (2) ImageProcessing class. The RGBImage class enabled us to initialize new instances of images in RGB color spaces. The ImageProcessing class enabled us to execute changes to the RGBImage instances.

These are the original pictures that we had:

A red and white logo

Description automatically generated with low confidenceBackground pattern

Description automatically generated

After executing the rotate 180 degrees method, this image is created:

A picture containing text, clipart, vector graphics

Description automatically generated

A red and white logo

Description automatically generated with medium confidenceAfter executing the crop method, this image is created:

After executing the negate method, this image is created:

A picture containing qr code

Description automatically generated

Qr code

Description automatically generated with medium confidenceAfter executing the chroma key (overlap) method with the original two images, this image is created:

After executing the tint method, this image is created:

A picture containing qr code

Description automatically generated

After executing the clear channel method (clearing two different intensity channels), these images are created:

A picture containing qr code

Description automatically generatedA picture containing qr code

Description automatically generated

Part 2: Modified KNN classifier-

We created a simple KNN classifier (ImageKNNClassifier class) that takes in training data (pictures with their labels) and a test image to predict the label of the test image.

We created random images with “low” and “high” intensity values as the training data. Then, we created low and high intensity images and tested to see if our KNN classifier worked as intended:

Text

Description automatically generated