**Image Processing Program: A Practical Approach of Linear Algebra in Google Colab**

By To

Asif Tauhid (st4787) Prof. Joachim Lebovits

Komal Neupane (kn2200)

**Introduction**

This report describes a simple image processing program implemented in Google Colab that demonstrates some fundamental operations such as cropping, resizing, and rotating images. The program utilizes basic concepts of linear algebra to manipulate image data, providing a practical demonstration of how mathematical concepts are directly applied in computer science and image processing.

**Program Overview**

The program is designed to run in a Google Colab notebook, leveraging minimal Python libraries to handle image input/output while focusing on manual implementations of image transformations. This approach highlights the application of linear algebra in manipulating the array-based data structures that represent images.

**Key Features and Functions**

**1. Image Loading**

The program begins by loading an image from a specified path in Google Drive. This step uses the imageio library, which simplifies reading the image into a numpy array. Each pixel in an image is represented as a combination of RGB (red, green, blue) values, stored in a matrix form (a two-dimensional array).

**2. Cropping**

Cropping is achieved by selecting a submatrix from the main matrix representing the image. This is done by specifying start and end points along the rows and columns of the array. This process directly utilizes the concept of submatrices in linear algebra, where a portion of the original matrix (image) is selected and isolated.

**3. Resizing**

The resizing function changes the dimensions of the image, effectively altering the resolution. This is implemented by computing the ratio of new dimensions to old dimensions and then selecting pixels at intervals determined by these ratios. The function iterates over the desired output size and samples pixels from positions scaled according to these ratios. This operation can be seen as a practical application of matrix transformation, where the original matrix is transformed into a new matrix of the desired size through a process similar to matrix interpolation.

**4. Rotating**

Rotation of the image by 90, 180, or 270 degrees is performed using numpy's rot90 function, which is a direct application of matrix rotation in linear algebra. The function rearranges the matrix elements to rotate the image, which is equivalent to performing a linear transformation on the matrix representing the image.

**5. Displaying Images**

After each transformation, the image is displayed in the notebook. This allows for immediate visualization after each of the matrix operations are performed on the image data.

**Linear Algebra Connections**

Linear algebra provides the mathematical foundation for operations on matrices, which are crutial to image processing tasks. Each pixel of an image can be thought of as a vector, and the entire image as a matrix of vectors. Operations such as cropping, resizing, and rotating involve manipulating these matrices:

* Cropping involves extracting a submatrix based on specific row and column indices.
* Resizing transforms the matrix to a new size, maintaining as much of the original image's layout as possible, which is known as transformations in linear algebra.
* And at the end, rotating the image involves matrix transposition and reorientation.

Overall, this program demonstrates how linear algebra is not just a field of abstract mathematical theory but a practical toolkit for modern computer applications like image processing. By manipulating the matrices that represent digital images, we can perform complex transformations that are both foundational and essential for advanced image processing tasks, computer graphics, and many other applications in technology and science.