Basic Image Processing Using JAVA

01.09.2023

—

Sai Venkatesh Alampally
Batch - 2
sai.23bcs10095@ms.sst.scaler.com
+91 8074962223

Overview

This document presents a JAVA-based Image Editor developed by Sai Venkatesh as part of a mini-project for Scaler School of Technology's Computer Programming class.This project offers various options to user like Rotating Image,Converting Image to grayscale, blurring the Image e.t.c…, all accessible through a text-based interface.

Problem Statement

1. This project demands me to think about the need of a flexible image processing tool that can be used for multiple enhancements in an image.The application should be able to load an image from a file, apply different image manipulation techniques, and save the edited image back to a file.
2. The Goal is to provide a user-friendly interface for performing basic Image Editing Tasks

Code Structure

The code consists of different functions and also a Main class (Main.java) which controls all functionalities.

Available Functions :

```
VERTICAL Inversion
HORIZONTAL Inversion
Rotate Left
Rotate Right
Print PIXEL VALUES of image
Change IMAGE BRIGHTNESS
convert image to GRAYSCALE
convert image to GREENSCALE
convert image to BLUESCALE
convert image to REDSCALE
```

Implementation and Result

I. Vertical Inversion

```
    public static BufferedImage verticallyinvert(BufferedImage inImage) {
```

```
        int height = inImage.getHeight();
        int width = inImage.getWidth();
        BufferedImage VinvertImage = new BufferedImage(width, height,
    BufferedImage.TYPE_INT_RGB);
        for (int j = 0; j < width; j++) {
            for (int i = 0; i < height; i++) {

                VinvertImage.setRGB(j, height - i - 1, inImage.getRGB(j,i));
            }
        }

        return VinvertImage;
    }
```

Reference Image

II. Horizontal Inversion

```
    public static BufferedImage horizontallyinvert(BufferedImage inImage) {
        int height = inImage.getHeight();
        int width = inImage.getWidth();
        BufferedImage HinvertImage = new BufferedImage(width, height,
    BufferedImage.TYPE_INT_RGB);
        for (int i = 0; i < height; i++) {
            for (int j = 0; j < width; j++) {
                HinvertImage.setRGB(width - j - 1, i, inImage.getRGB(j, i));
            }
```

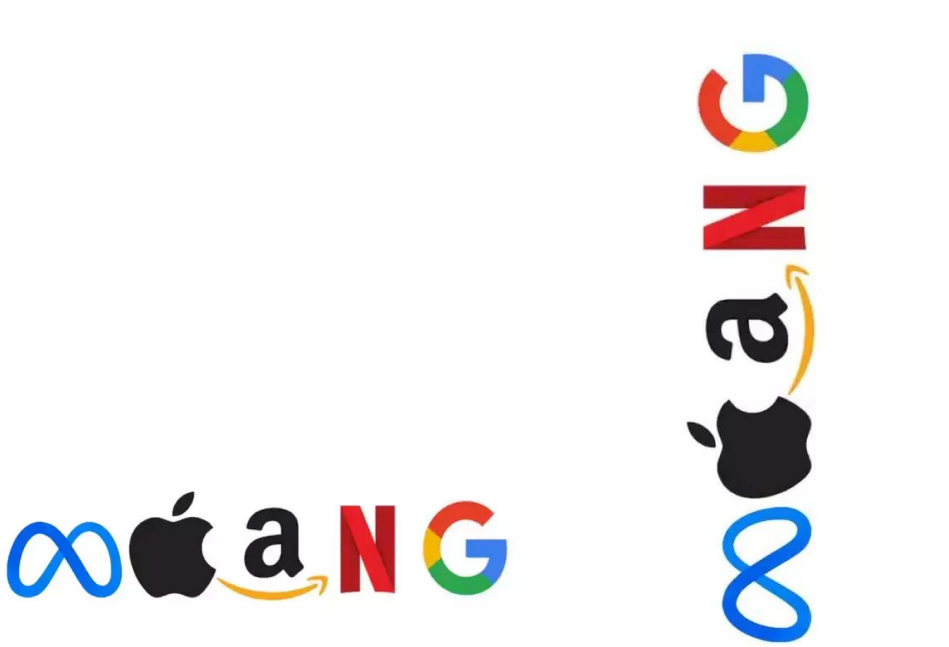
```
        }
            return HinvertImage;
    }
```

Reference Image

III. Rotate 90 degrees Anti-Clockwise

```
    public static BufferedImage Lefttransposeimage(BufferedImage inImage) {
        int height = inImage.getHeight();
        int width = inImage.getWidth();
        BufferedImage rotatedImage = new BufferedImage(height, width,
    BufferedImage.TYPE_INT_RGB);
        for (int i = 0; i < height; i++) {
            for (int j = 0; j < width; j++) {
                rotatedImage.setRGB(i,j,inImage.getRGB(j,i));
            }
        }
        int index=rotatedImage.getHeight()-1;
        for(int j=0;j<rotatedImage.getWidth();j++){
            for(int i=0;i<rotatedImage.getHeight()/2;i++){
                Color temp=new Color(rotatedImage.getRGB(j,i));
                rotatedImage.setRGB(j,i,rotatedImage.getRGB(j,index-i));
                rotatedImage.setRGB(j,index-i,temp.getRGB());
            }
        }
        return rotatedImage;
    }
```

Reference Image



IV. Rotate 90 degrees Clockwise

```
public static BufferedImage Righttransposeimage(BufferedImage inImage) {
    int height = inImage.getHeight();
    int width = inImage.getWidth();
    BufferedImage rotatedImage = new BufferedImage(height, width, BufferedImage.TYPE_INT_RGB);
    for (int i = 0; i < height; i++) {
        for (int j = 0; j < width; j++) {
            rotatedImage.setRGB(i,j,inImage.getRGB(j,i));
        }
    }
    int index=rotatedImage.getWidth()-1;
    for(int i=0;i<rotatedImage.getHeight();i++){
        for(int j=0;j<rotatedImage.getWidth()/2;j++){
            Color temp=new Color(rotatedImage.getRGB(j,i));
            rotatedImage.setRGB(j,i,rotatedImage.getRGB( x: index-j,i));
            rotatedImage.setRGB( x: index-j,i,temp.getRGB());
        }
    }
    return rotatedImage;
}
```

Reference Image

V. Change Image brightness

```
public static BufferedImage changebrightness(BufferedImage inImage,int a){
    int height = inImage.getHeight();
    int width = inImage.getWidth();
    BufferedImage output = new BufferedImage(width,height,BufferedImage.TYPE_3BYTE_BGR);
    for(int i=0;i<height;i++){
        for(int j=0;j<width;j++){
            Color pixel = new Color(inImage.getRGB(j,i));
            int red = pixel.getRed();
            int green = pixel.getGreen();
            int blue = pixel.getBlue();
            red+= (a*red)/100;
            green+=(a*green)/100;
            blue+=(a*blue)/100;
            if(red>255)
                red=255;
            if(green>255)
                green=255;
            if(blue>255)
                blue=255;
            if(red<0)
                red=0;
            if(green<0)
                green=0;
            if(blue<0)
                blue=0;
            Color newpixel = new Color(red,green,blue);
            inImage.setRGB(j,i,newpixel.getRGB());}}
    return inImage;
}
```

Reference Image

VI. Convert Image To Gray Scale

```java
public static BufferedImage convertToGrayScale(BufferedImage inputImage) {
    int height = inputImage.getHeight();
    int width = inputImage.getWidth();
    BufferedImage outputImage = new BufferedImage(width, height, BufferedImage.TYPE_BYTE_GRAY);
    for (int i = 0; i < height; i++) {
        for (int j = 0; j < width; j++) {
            outputImage.setRGB(j, i, inputImage.getRGB(j, i));
        }
    }
    return outputImage;
}
```

Reference Image

VII. Convert Image To Green Scale

```
public static BufferedImage GreenScale(BufferedImage inImage){
    int height = inImage.getHeight();
    int width = inImage.getWidth();
    BufferedImage output = new BufferedImage(width,height,BufferedImage.TYPE_3BYTE_BGR);
    for(int i=0;i<height;i++){
        for(int j=0;j<width;j++){
            Color pixel = new Color(inImage.getRGB(j,i));
            int red = pixel.getRed();
            int green = pixel.getGreen();
            int blue = pixel.getBlue();
            green+=(100*green)/100;
            if(green>255)
                green=255;
            if(green<0)
                green=0;
            Color newpixel = new Color(red,green,blue);
            output.setRGB(j,i,newpixel.getRGB());}}
    return output;
}
```

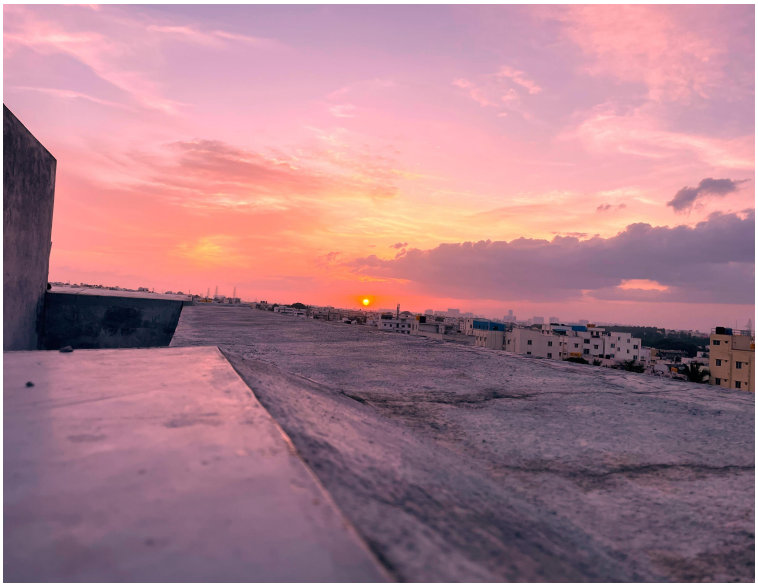
Reference Image

VIII. Convert Image To Red Scale

```
public static BufferedImage RedScale(BufferedImage inImage){
    int height = inImage.getHeight();
    int width = inImage.getWidth();
    BufferedImage output = new BufferedImage(width,height,BufferedImage.TYPE_3BYTE_BGR);
    for(int i=0;i<height;i++){
        for(int j=0;j<width;j++){
            Color pixel = new Color(inImage.getRGB(j,i));
            int red = pixel.getRed();
            int green = pixel.getGreen();
            int blue = pixel.getBlue();
            red+= (50*red)/100;
            if(red>255)
                red=255;
            if(red<0)
                red=0;
            Color newpixel = new Color(red,green,blue);
            inImage.setRGB(j,i,newpixel.getRGB());}}
    return inImage;


}
```

Reference Image



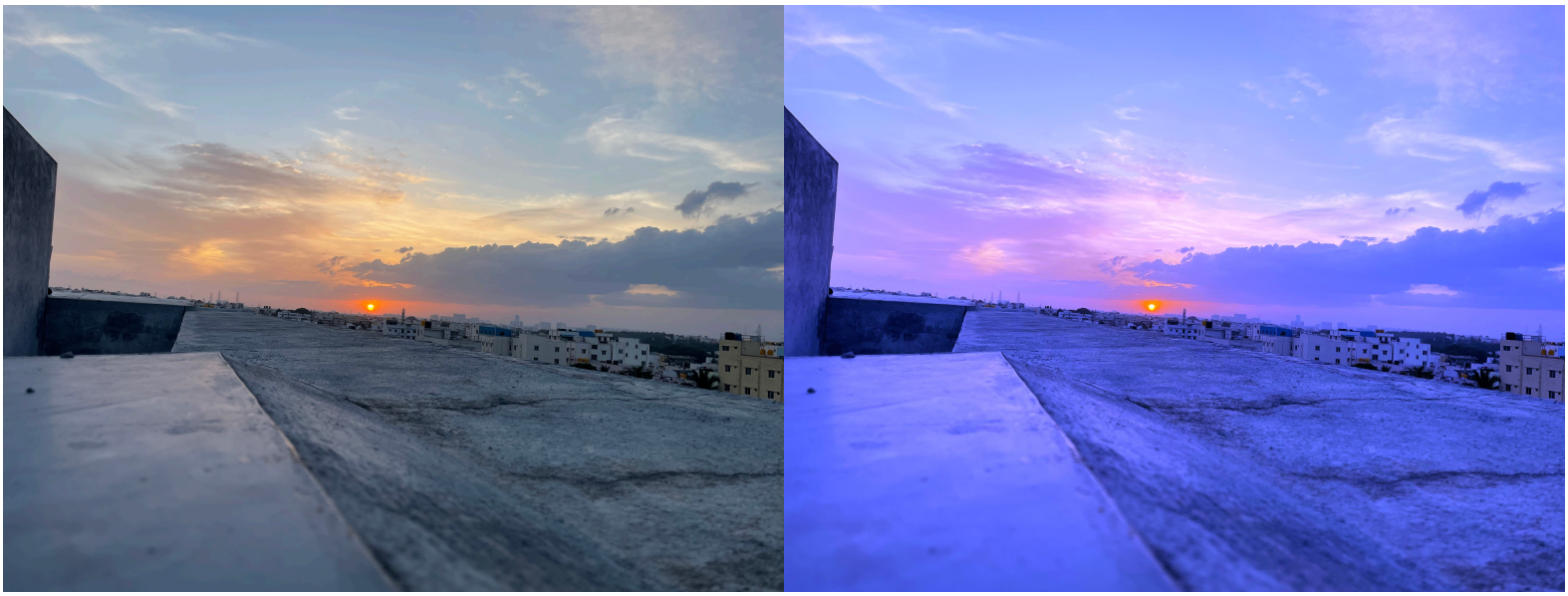
IX. Convert Image To Blue Scale

```
public static BufferedImage BlueScale(BufferedImage inImage){
    int height = inImage.getHeight();
    int width = inImage.getWidth();
    BufferedImage output = new BufferedImage(width,height,BufferedImage.TYPE_3BYTE_BGR);
    for(int i=0;i<height;i++){
        for(int j=0;j<width;j++){
            Color pixel = new Color(inImage.getRGB(j,i));
            int red = pixel.getRed();
            int green = pixel.getGreen();
            int blue = pixel.getBlue();
            blue+=(100*blue)/100;

            if(blue>255)
                blue=255;

            if(blue<0)
                blue=0;
            Color newpixel = new Color(red,green,blue);
            inImage.setRGB(j,i,newpixel.getRGB());}}
    return inImage;
}
```

Reference Image



X. Blur Image

```java
public static BufferedImage blurImage(BufferedImage inImage,int blrRatio){
    int height = inImage.getHeight();
    int width = inImage.getWidth();
    BufferedImage outputImage = new BufferedImage(width, height , BufferedImage.TYPE_INT_RGB);
    for (int i=0;i<height;i+=blrRatio){
        for (int j=0;j<width;j+=blrRatio){
            int sumofRED=0,sumofGREEN=0,sumofBLUE=0,sumofAlpha=0;
            int count=0;
            for(int k=i;k<i+blrRatio;k++){
                for (int l=j;l<j+blrRatio;l++){
                    if (k<height && l<width) {
                        Color pixel = new Color(inImage.getRGB(l, k));
                        sumofRED += pixel.getRed();
                        sumofGREEN += pixel.getGreen();
                        sumofBLUE += pixel.getBlue();
                        sumofAlpha += pixel.getAlpha();
                        count++;
                    }
                }
            }
            sumofRED=sumofRED/count;
            sumofGREEN=sumofGREEN/count;
            sumofBLUE=sumofBLUE/count;
            sumofAlpha=sumofAlpha/count;
            for(int k=i;k<i+blrRatio;k++){
                for(int l=j;l<j+blrRatio;l++){
                    if (k<height && l<width){
                        Color newpixel = new Color(sumofRED,sumofGREEN,sumofBLUE,sumofAlpha);
                        outputImage.setRGB(l,k,newpixel.getRGB());
                    }
                }
            }

        }
    }
    return outputImage;
}
```

Reference Image

Conclusion

In conclusion, the Image Processing Application provides a user-friendly interface for performing various image processing tasks. It offers versatility and ease of use, making it suitable for both novice and experienced users. The project demonstrates the capability of Java for image processing tasks and lays the foundation for potential enhancements and future improvements.

Acknowledgement

I would like to extend my sincere thanks to the instructor Kshitij Mishra for giving us such a wonderful opportunity to work on this project ,which helped me to revise my concepts and learnt to implement the learnings to solve a real life problem .

I would also like to express my gratitudes to the BSM Diwakar Gupta for providing invaluable guidance that made this project possible.

His contributions and collaborative spirit greatly enriched this learning experience.