

Term Project Report

```
for(int xx = 0; xx<w;xx++){  
    for(int yy = 0; yy<h;yy++){  
        int pixel = tempimage.getRGB(xx, yy);  
        int red = (pixel >> 16) & 0x0ff;  
        int green = (pixel >> 8) & 0x0ff;  
        int blue = (pixel) & 0x0ff;  
        int R = red, G = green, B = blue;  
        tempimage.setRGB(xx, yy, new Color(R, G, B).getRGB());  
    }  
}
```

In for loop **xx** takes each pixel from 0 to the width of image, **yy** takes each pixel from 0 to height of image. It takes red, green, blue values of each pixel. For example it takes red, green, blue value of **(0,0)** coordinate.

In our first if statement

```
if(bp.getjCheckBox1()){  
    R = (int) Math.sqrt(Math.pow(blue, 2) + red) /2;  
    B = (int) Math.sqrt(Math.pow(red, 2) + Math.sqrt(blue))/2;  
    G = (int) Math.sqrt(Math.pow(red, 2) + Math.sqrt(green))/2;  
}
```

if blue tone of image is more, in this case red tones of an image will increase



Before applying
effect



After applying
effect

If red tone of image is more, then green and blue tones of an image will increase



Before applying
effect



After applying
effect

In second if statement

```
if(bp.getjCheckBox2()){  
    R = red;  
    G = (int) (Math.sqrt(green));  
    B = (int) (Math.sqrt(blue));  
}
```

This effect keep red tone of image and decreasing green and blue value of image.



Before applying
effect



After applying
effect

```
if(bp.getjCheckBox3()){  
    R = blue;  
    G = red;  
    B = green;  
}
```



In our image, especially on car red tones is more than others. According to my code that "G = red", green value of car will increase, like in the second image. In the sky, blue tones is more, therefore sky will red tone, because "R = blue".

```
if(bp.getjCheckBox4()){  
    double max = 0;  
    double min = 0;  
    max = Math.max(red, blue);  
    min = Math.min(blue, green);  
    if(max == red){  
        R = blue;  
    }  
    if(min == green){  
        B = green;  
    }else{  
        G = blue;  
    }  
}
```

In the 4th if statement firstly find maximum value between red and blue. If red is more than blue in selected pixel, then red channel will equal to blue channel's value.

Second step find minimum value between blue and green. If min value is equal to green, then blue channel will equal to green in selected pixel. Otherwise green channel value will equal to blue channel's value.

Blur effect

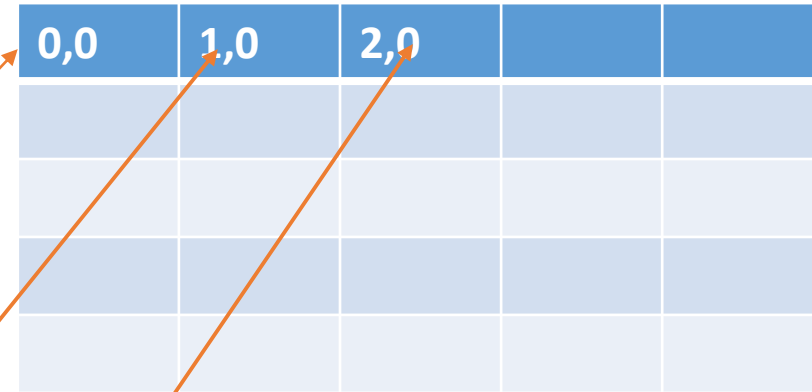
```
if(bp.getjCheckBox5()){  
    for (int xx = 1; xx < w-1; xx++){  
        for (int yy = 1; yy < h-1; yy++) {
```

```
            int R = 0;  
            int G = 0;  
            int B = 0;
```

```
            int pixel1 = tempimage.getRGB(xx - 1, yy - 1);  
            R += (pixel1 >> 16) & 0xff;  
            G += (pixel1 >> 8) & 0xff;  
            B += (pixel1 >> 0) & 0xff;
```

```
            int pixel2 = tempimage.getRGB(xx, yy - 1);  
            R += (pixel2 >> 16) & 0xff;  
            G += (pixel2 >> 8) & 0xff;  
            B += (pixel2 >> 0) & 0xff;
```

```
            int pixel3 = tempimage.getRGB(xx + 1, yy - 1);  
            R += (pixel3 >> 16) & 0xff;  
            G += (pixel3 >> 8) & 0xff;  
            B += (pixel3 >> 0) & 0xff;
```



0,0	1,0	2,0		

Here I create 3x3
matrix from image
pixels.

```
int pixel4 = tempimage.getRGB(xx - 1, yy);
```

```
R += (pixel4 >> 16) & 0xff;
```

```
G += (pixel4 >> 8) & 0xff;
```

```
B += (pixel4 >> 0) & 0xff;
```

```
int pixel5 = tempimage.getRGB(xx, yy);
```

```
R += (pixel5 >> 16) & 0xff;
```

```
G += (pixel5 >> 8) & 0xff;
```

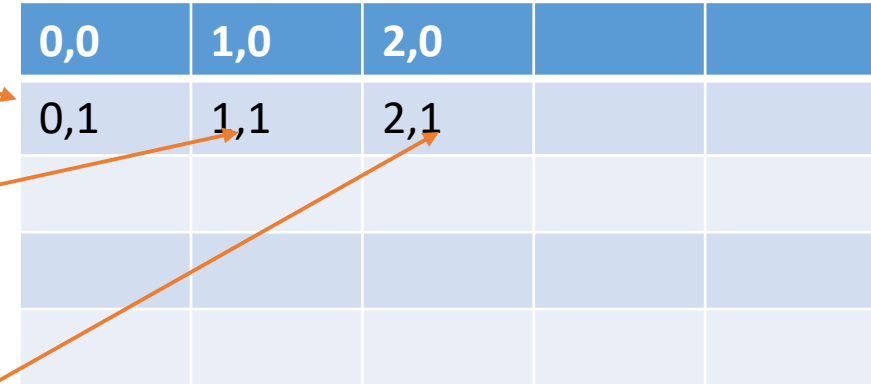
```
B += (pixel5 >> 0) & 0xff;
```

```
int pixel6 = tempimage.getRGB(xx + 1, yy);
```

```
R += (pixel6 >> 16) & 0xff;
```

```
G += (pixel6 >> 8) & 0xff;
```

```
B += (pixel6 >> 0) & 0xff;
```

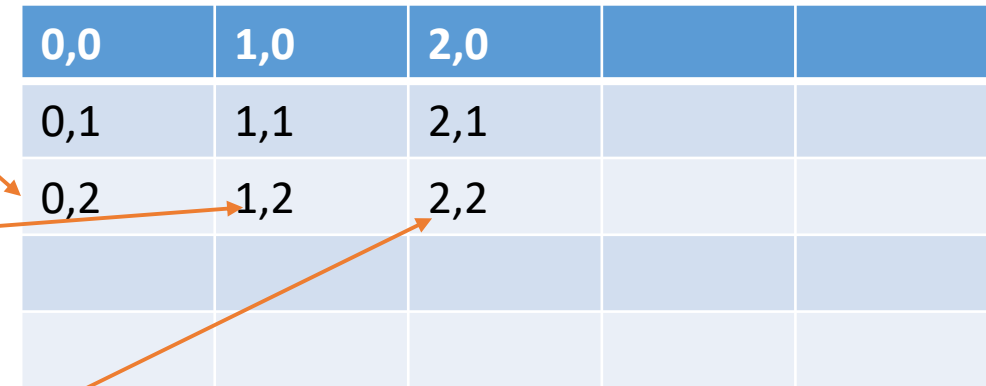


0,0	1,0	2,0		
0,1	1,1	2,1		


```
int pixel7 = tempimage.getRGB(xx - 1, yy + 1);  
R += (pixel7 >> 16) & 0xff;  
G += (pixel7 >> 8) & 0xff;  
B += (pixel7 >> 0) & 0xff;
```

```
int pixel8 = tempimage.getRGB(xx, yy + 1);  
R += (pixel8 >> 16) & 0xff;  
G += (pixel8 >> 8) & 0xff;  
B += (pixel8 >> 0) & 0xff;
```

```
int pixel9 = tempimage.getRGB(xx + 1, yy + 1);  
R += (pixel9 >> 16) & 0xff;  
G += (pixel9 >> 8) & 0xff;  
B += (pixel9 >> 0) & 0xff;
```

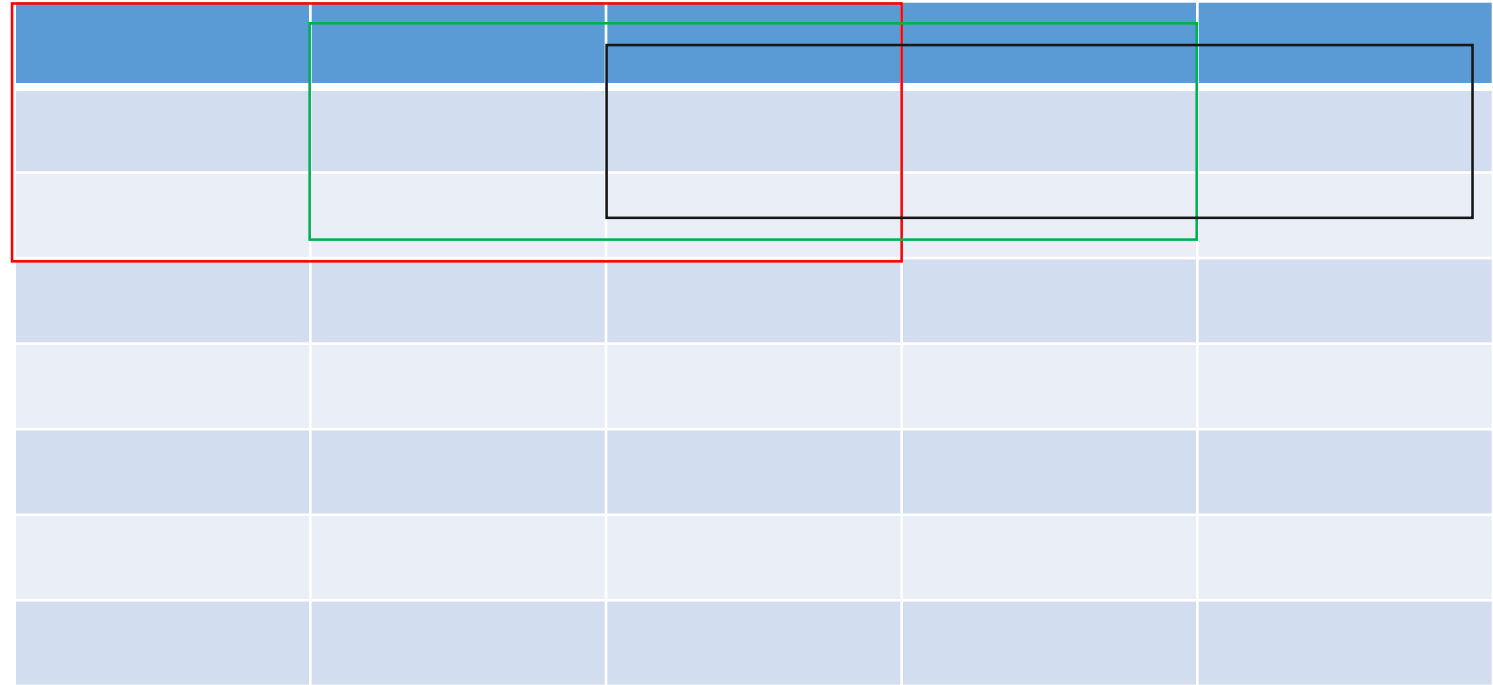


0,0	1,0	2,0		
0,1	1,1	2,1		
0,2	1,2	2,2		

```
R /= 9;      This statement  
G /= 9;      normalize Red, Green  
B /= 9;      and Blue colors value.
```

3x3 matrix loop through
full image.

Assume that it is image



```
tempimage.setRGB(xx, yy, new Color(R,G,B).getRGB());
```

This code assign new values into
each pixel.

Sobel Filter

```
int pixel1 = getGrayScale(tempimage.getRGB(xx - 1, yy - 1));  
int pixel2 = getGrayScale(tempimage.getRGB(xx - 1, yy));  
int pixel3 = getGrayScale(tempimage.getRGB(xx - 1, yy + 1));
```

```
int pixel4 = getGrayScale(tempimage.getRGB(xx, yy - 1));  
int pixel5 = getGrayScale(tempimage.getRGB(xx, yy));  
int pixel6 = getGrayScale(tempimage.getRGB(xx, yy + 1));
```

```
int pixel7 = getGrayScale(tempimage.getRGB(xx + 1, yy - 1));  
int pixel8 = getGrayScale(tempimage.getRGB(xx + 1, yy));  
int pixel9 = getGrayScale(tempimage.getRGB(xx + 1, yy + 1));
```

This code take 3x3 matrix from image and loop through full image.
 $G = \sqrt{G_x^2 + G_y^2}$ according to this equation we will find G_x and G_y .

```
double gx = (pixel1 * 1 * pow) + (pixel2 * 2 * pow)
            + (pixel3 * 1 * pow)
            +(pixel7 * -1 * pow) + (pixel8 * -2 * pow)
            + (pixel9 * -1 * pow);
```

```
double gy = (pixel1 * 1 * pow) + (pixel3 * -1 * pow)
            +(pixel4 * 2 * pow) + (pixel6 * -2 * pow)
            +(pixel7 * 1 * pow) + (pixel9 * -1 * pow);
```


After finding gx and gy , write gx and gy
in equation that in the before slide.

```
double gval = Math.sqrt(Math.pow(gx, 2)+ Math.pow(gy, 2));
int val = (int) gval;
```

After that operation copy
result into new matrix.

```
edgeColors[xx][yy] = val;
```

```
for (int i = 0; i < w; i++) {  
    for (int j = 0; j < h; j++) {  
        int edgeColor = edgeColors[i][j];  
        edgeColor = (int)(edgeColor );  
        edgeColor = (edgeColor << 16) | (edgeColor << 8) |  
edgeColor;  
  
        tempimage.setRGB(i, j, edgeColor);  
    }  
}
```



Create image from
new matrix.

Brightness

```
int bRed = (red+red*bright/100)>255 ? 255 : (red+red*bright/100)<0 ? 0 : (red+red*bright/100);  
int bGreen = (green+green*bright/100)>255 ? 255 : (green+green*bright/100)<0 ? 0 : (green+green*bright/100);  
int bBlue = (blue+blue*bright/100)>255 ? 255 : (blue+blue*bright/100)<0 ? 0 : (blue+blue*bright/100);
```

Bright variable takes value from slider and apply it in the above code

Save Process

```
public void save(File file){  
    imagePanel1.save(file);  
}
```

In the main panel , I write a method that have file parameter. In this method I pass file argument into save function in the image panel.

```
public void save(File file){  
    if(file!=null){  
        try{  
            file = new File(file + "-edit.png");  
            ImageIO.write(tempimage, "png", file);  
            JOptionPane.showMessageDialog(this, "Your changes saved");  
  
        }catch(IOException ex){  
            JOptionPane.showMessageDialog(this, "Something went wrong");  
        }  
    }else{  
        JOptionPane.showMessageDialog(this, "No image selected yet");  
    }  
}
```

This function save last version of image.