

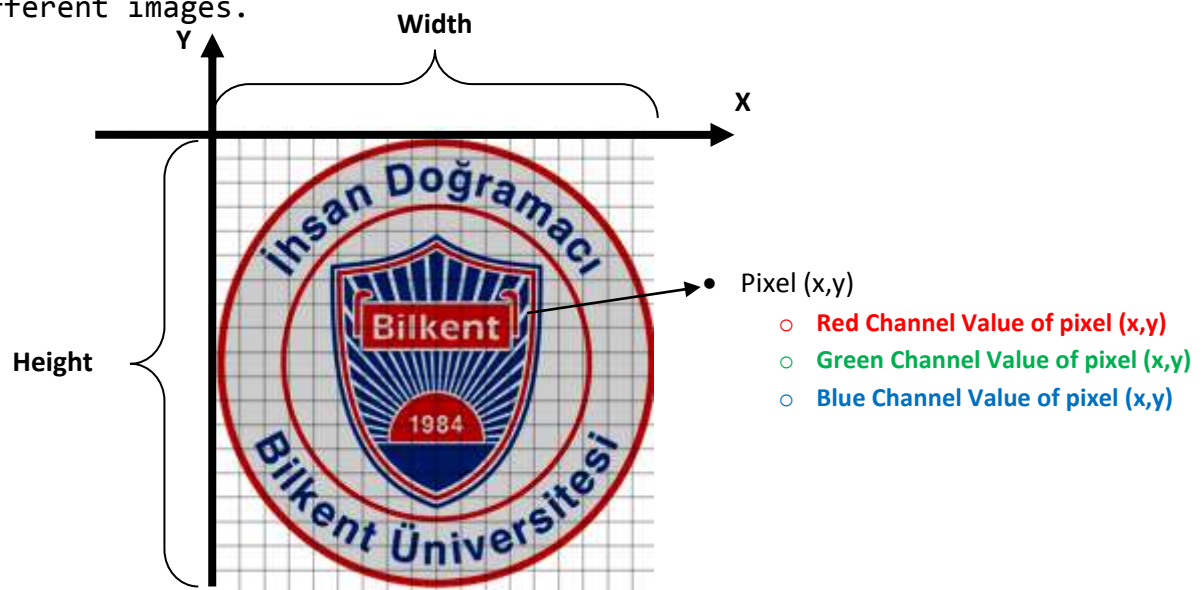
## LAB EXAM 9

### Section 2

Apr. 19, 2016

In this lab assignment, you are going to write a program that applies gamma and brightness correction operations on an image.

Basically, an image is a 2D array that holds RGB (red, green, blue) values at each array position, which is called pixel. From an RGB value, the red, green and blue values can be obtained separately, which are called, red channel, green channel and blue channel. Main motivation of image processing is to manipulate channel values and obtain different images.



First of all, an Image class is given to you, which performs basic Image I/O (input / output) and channel (red, green, blue) operations. So, you will not need to implement it. Information about Image class is given below;

1. To load an image simply create an instance of Image class by giving its file name (i.e. Picture.jpg) of the image as an parameter;

```
Image orgPicture= new Image("Picture.jpg");
```

2. You can get width and height of an image by;

```
myPicture.getWidth()  
myPicture.getHeight()
```

3. After loading the image, you can get channel values which are 2D integer arrays;

```
myPicture.getRedChannel(); // returns int[][]  
myPicture.getGreenChannel(); // returns int[][]  
myPicture.getBlueChannel(); // returns int[][]
```

4. You can create a new image with channel values arrays.

```
Image modifiedImage = new Image(int[][] red, int[][] green, int [][] blue)
```

5. Additionally, you can display the created image by;  
`manipulatedPicture.display()`

You will write an ImageProccesing class which creates an image from a path (Above by using step 1), gets the channel values (step 2), performs gamma correction and brightness correction operations on the image (you will implement), and displays manipulated images.

## Brightness Correction Algorithm:

The user will be asked for a brightness value for the operation, then your program should calculate new red, green, blue channels for the result image. Keep in mind that, you have to calculate new channel values for all pixels.

**Resulting image's red channel value at pixel (x,y) equals**

`truncate (original image's red channel value at pixel(x,y) + brightness)`

**Resulting image's green channel value at pixel (x,y) equals:**

`truncate (original image's green channel value at pixel(x,y) + brightness)`

**Resulting image's blue channel value at pixel (x,y) equals:**

`truncate (original image's blue channel value at pixel(x,y) + brightness)`

You have to write the `truncate (int arg)` method that returns 0 if the arg < 0 and returns 255 if the arg > 255.

After obtaining channel values for the result image; you should create an instance of Image by using step 4 and display it.

## Gamma Correction Algorithm:

First specify a gamma value for the operation then calculate new red, green, blue channels for the result image. Keep in mind that, you have to calculate new channel values for all pixels.

`gammaCorrection = 1/gamma`

**Result image's red channel value at pixel (x,y) equals**

`255 * (original image's red channel value at pixel(x,y) / 255))gammaCorrection`

**Result image's green channel value at pixel (x,y) equals:**

`255 * (original image's green channel value at pixel(x,y) / 255))gammaCorrection`

**Result image's blue channel value at pixel (x,y) equals:**

`255 * (original image's blue channel value at pixel(x,y) / 255))gammaCorrection`

After obtaining channel values for the result image; you should create an instance of Image by using step 4 and display it.

ImageProcessing class should contain following methods;

- correctBrightness, which gets a brightness value and an image as an input and returns corrected image object.
- correctGamma, which gets a gamma value and an image as an input and returns corrected image object.
- truncate, which gets a double value and returns a truncated value as discussed above.

**Sample run :**

Enter the name of the file: Car.jpg

Enter brightness: -60

Enter gamma: 4

Original image:



After correcting brightness with -60 (darkening)



After correcting gamma with 4



Note that this is an exam. You are not allowed to communicate with any person other than your teaching assistant. Those who do not obey this rule will be subject to disciplinary investigation.

You can use only the features (techniques, classes, methods and statements) that are covered in the class. After finishing your work, select your folder (e.g. Lab09), then right click and select "Send to" option. Click "Compressed (zipped) folder". Rename your zip file as "Lab09\_Surname\_Name". Note that, your file type is "zip", namely your file name will not be "Lab09\_Surname\_Name.zip". Upload your zip file to Unilica. Note that you must upload your file before 16:40; you cannot upload your file after that time.

## // Image.java

```
import java.awt.Color;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JLabel;

public class Image {

    private BufferedImage img;
    private int height;
    private int width;

    public Image (String filePath) {
        img = null;
        try {
            img = ImageIO.read(new File(filePath));
            height = img.getHeight();
            width = img.getWidth();
        }
        catch (IOException e) {}
    } // Image

    public Image(int[][] r, int[][] g, int [][] b) {

        width= r.length;
        height=r[0].length;

        img = new BufferedImage(width,height, BufferedImage.TYPE_INT_ARGB);

        for (int i = 0; i < width; i++) {
            for (int j = 0; j < height; j++) {
                Color c = new Color (r[i][j],g[i][j],b[i][j]);
                img.setRGB(i, j, c.getRGB());
            } // for
        } // for
    } // Image

    public int[][] getRedChannel() {
        int[][] redPixels = new int[width][height];

        for( int i = 0; i < width; i++ ){
            for( int j = 0; j < height; j++ ) {
                Color c = new Color(img.getRGB(i,j));
                redPixels[i][j] = c.getRed();
            } //for
        } //for
        return redPixels;
    } //getRedChannel

    public int[][] getGreenChannel() {
        int[][] greenPixels = new int[width][height];

        for( int i = 0; i < width; i++ ) {
            for( int j = 0; j < height; j++ ) {
                Color c = new Color(img.getRGB(i,j));
                greenPixels[i][j] = c.getGreen();
            } //for
        } //for
        return greenPixels;
    } //getGreenChannel

    public int[][] getBlueChannel() {
        int[][] bluePixels = new int[width][height];

        for( int i = 0; i < width; i++ ){
            for( int j = 0; j < height; j++ ) {
                Color c = new Color(img.getRGB(i,j));
                bluePixels[i][j] = c.getBlue();
            } //for
        } //for
    } //getBlueChannel
}
```

```
        } //for
        return bluePixels;
    } // getBlueChannel

    public void display() {
        JFrame frame = new JFrame("WINDOW");
        frame.setVisible(true);
        frame.setSize(width, height);
        frame.add(new JLabel(new ImageIcon(this.img)));
    } // display

    public int getHeight() {
        return height;
    }

    public int getWidth() {
        return width;
    }
} // class Image
```