|  |  |
| --- | --- |
| **AP Computer Science I** | **Image Processing Part 1 GrayScaling Lab** |
| **GrayScaling Program** | **100 Point Version** |
| **Assignment Purpose:**  In this assignment you are to read in an image and use the APImage class and the Pixel class to convert the color image into a grayscale image. You will not need to save the image, just display it in a window. | |

**Getting Started:**

You will need to copy the image folder, the provided image (or use one of your own choosing) and Student\_Grayscale.java file to your drive space to get started.

The **algorithm** you will need to follow is shown below:

For every pixel in the image

Get the pixel’s red value

Get the pixel’s green value

Get the pixel’s blue value

Compute the average of the red, green and blue values

Set the red to the average

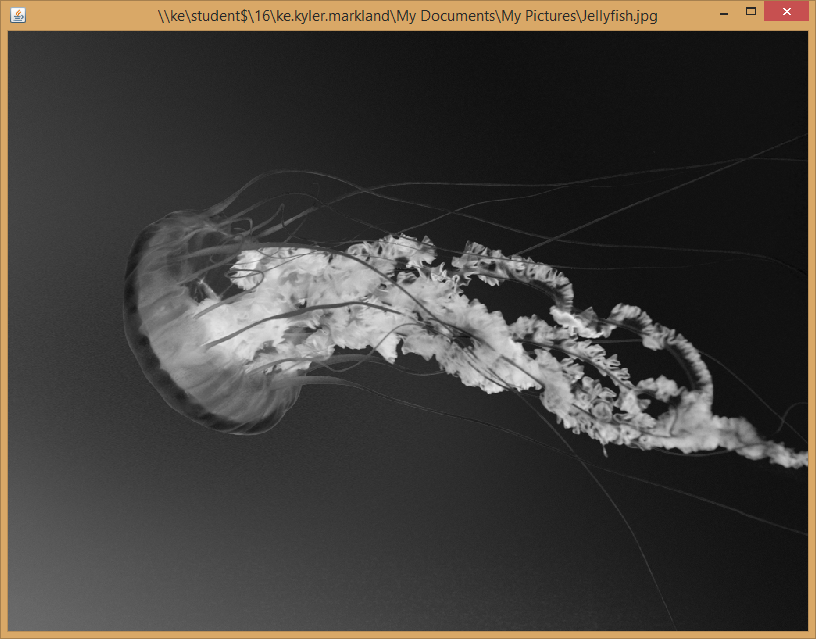
Set the green to the average

Set the blue to the average

Display the image

**100 Point Version:**

Your result should appear similar to the following:



Here is a copy of the program stub that will get you started.

**import** javax.swing.\*;

**import** images.\*; //imports APImages class

**import** java.io.\*;

**public** **class** Grayscale

{

**public** **static** **void** main(String args[])

{

JFileChooser fc = **new** JFileChooser();

fc.showOpenDialog(**null**);

File file = fc.getSelectedFile();

**if** (file == **null**)

System.*exit*(0);

APImage img = **new** APImage(file.getPath());

//Insert code for grayscale conversion here

img.draw();

}

}

**The Algorithm**

For every pixel in the image

Get the red value for the pixel and keep a red total

Get the green value for the pixel and keep a green total

Get the blue value for the pixel and keep a blue total

Read in the other 8 pixels

Get the red value for the pixel and add to red total

Get the green value for the pixel and add to green total

Get the blue value for the pixel and add to blue total

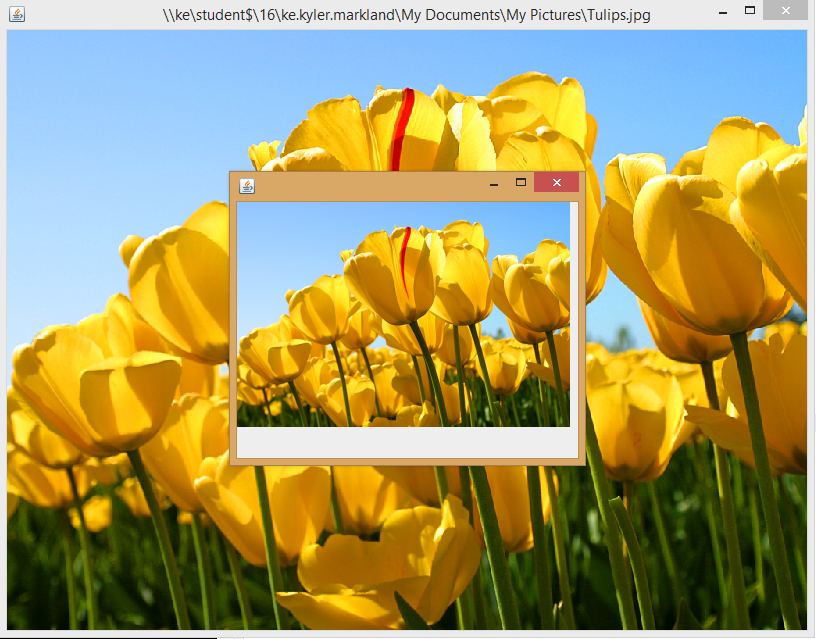
Average red values

Average green values

Average blue values

Set the new pixel values for the resized image.

|  |  |
| --- | --- |
| **AP Computer Science I** | **Image Processing Part 2 Resizing Lab** |
| **Resizing Program** | **100 Point Version** |
| **Assignment Purpose:**  In this assignment you will read in an image and use the APImage and Pixel classes to resize and display the original image and the resized image in new windows. | |



Here is a copy of the program stub that will get you started.

**import** javax.swing.\*;

**import** images.\*;

**import** java.io.\*;

**public** **class** Resize

{

**public** **static** **void** main(String args[])

{

JFileChooser fc = **new** JFileChooser();

fc.showOpenDialog(**null**);

File file = fc.getSelectedFile();

**if** (file == **null**)

System.*exit*(0);

APImage img = **new** APImage(file.getPath());

APImage resizedImage = **new** APImage(img.getWidth() / 3, img.getHeight() / 3);

// Insert resizing code here

img.draw();

resizedImage.draw();

}

}

|  |  |
| --- | --- |
| **AP Computer Science I** | **Image Processing Part 3 Remove dust or speckles Lab** |
| **Despeckle Program** | **100 Point Version** |
| **Assignment Purpose:**  In this assignment you will read in an image with dust spots or speckles on it and remove them. This problem often occurs with older black and white photos that get scanned into a computer. You will want to ask the user to specify the size of the speck to remove. The image you display will remove all specks smaller than the desired value. Then you will display the image without the specks. Once again, you will use the APImage and Pixel classes to search for speckles. | |

Removing specks or dust from a photo is not an exact science. You may wish to test your code on both speck1.jpg and speck2.jpg. If you are not seeing the specks, you may want to open the photos in Photoshop and zoom in a bit.



Algorithm to Remove Specks

Get the size (diameter) of the speck to remove from the user

Get the tolerance level (amount the colors around your pixel may differ in color)

Repeat until the radius is small (you will need to define small – experiment)

For every pixel

Get the sum of every pixel’s red inside the given radius (diameter value from the user / 2)

Get the sum of every pixel’s green inside the given radius

Get the sum of every pixel’s blue inside the given radius

Compute the average for each RGB sum

if pixel’s inside averages (within radius) meet the tolerance requirements

Get the sum of every pixel’s red outside the given radius

Get the sum of every pixel’s green outside the given radius

Get the sum of every pixel’s blue outside the given radius

Compute the average for each RGB sum

Fill in the pixels within the radius of the current area with pixels of the average color

Decrease the size of the radius and run the process again

Display the image

Getting Started Code

**import** javax.swing.\*;

**import** images.\*;

**import** java.io.\*;

**public** **class** Despeckle

{

**public** **static** **void** main(String args[])

{

JFileChooser fc = **new** JFileChooser();

fc.showOpenDialog(**null**);

File file = fc.getSelectedFile();

**if** (file == **null**)

System.*exit*(0);

APImage img = **new** APImage(file.getPath());

// Insert despeckle code here

img.draw();

}

}