**A**

**PROJECT REPORT ON**

**Image Enhancement**

**SUBMITTED IN PARTIAL FULFILLMENT**

**FOR THE AWARD OF M.Sc. PART- II**

**(COMPUTER SCIENCE)**

**YEAR 2012 – 2013**

**-BY-**

**Miss. Arati Baban Satape**

**UNDER THE GUIDANCE OF**

**Miss. Arati S. Gavas**

**DEPARTMENT OF COMPUTER SCIENCE**

 **NAVKONKAN EDUCATION SOCIETY’S**

**DR.DATAR SCIENCE, DR.BEHERE ARTS &**

**SHRI PILUKAKA JOSHI COMMERCE COLLEGE**

**CHIPLUN-415605**

**(Affiliated To The University Of Mumbai)**



# D. B. J. COLLEGE, CHIPLUN

# DEPARTMENT OF COMPUTER SCIENCE



**(Post-Graduate Section)**

PROJECT CERTIFICATE

This is to certify that the project done at D.B.J. college, Chiplun

by **Miss.Arati Baban Satape** (Seat no : ) in partial

fulfillment for M.Sc. degree examination has been found satisfactory.

This report had not been submitted for any other examination and does

not form part of any other course undergone by the candidate.

**Mr. S. J. Nalawade**

**Incharge,**

**Department of Computer Science,**

**D.B.J.College, Chiplun.**

**GUIDE CERTIFICATE**



This is to certify that **Miss. Arati Baban Satape** from ***M.Sc.-II (Computer Science)*** of D.B.J. College, Chiplun has successfully completed her project work titled **‘Image Enhancement’** under my guidance and direction.

This is a bona-fide work carried out during the academic year 2012-13 towards partial fulfillment of the **M.Sc. Degree in Computer** **Science** course offered through honorable **University of** **Mumbai.**

Throughout the project, he was found sincere, regular and hard working.

Authorized Signature

Miss.Arati Gavas

(D.B.J.Collage, Chiplun)

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr No** | **Topic** | | | **Page No** |
| **1** | **Project Overview** | | |  |
|  | 1.1 Introduction | | |  |
|  | 1.2 Objective and Scope | | |  |
|  | 1.3 Application | | |  |
|  | 1.4 Hardware and Software Specification | | |  |
|  | 1.5 Feasibility Study  1.5.1 Operational Feasibility  1.5.2 Technical Feasibility  1.5.3 Economical Feasibility | | |  |
| **2** | **Literature Survey** | | |  |
| **3** | **Analysis and Design** | | |  |
|  | 3.1Java As a Programming Language | | |  |
| **4** | **Description of the Research Work** | | |  |
|  | 4.1 | | **Introduction to Basic Concept** |  |
|  | 4.1.1 | | What is Image? |  |
|  | 4.1.2 | | Why Digital Image Processing required? |  |
|  | 4.1.3 | | [Overview of Image Processing and Analysis](http://micro.magnet.fsu.edu/primer/digitalimaging/russ/overview.html) |  |
|  | 4.1.4 | | Working Of Function |  |
|  | 4.1.5 | | Mathematical and Logical Operators |  |
|  | 4.1.6 | | Filters in Image Processing |  |
|  | 4.1.7 | | Histogram |  |
|  |  | | |  |
| **5** | **Designs** | | |  |
|  | 5.1 | Class Diagram | |  |
| **6** | **Screen Layouts** | | |  |
|  | 6.1 Modules | | |  |
|  | 6.2 Validations | | |  |
| 7 | **Future Enhancement** | | |  |
| **8** | **Limitations** | | |  |
| **9** | **Conclusions** | | |  |
| **10** | **References** | | |  |

**ACKNOWLEDGEMENT**

Through words I cannot express my feeling but still here I grab this opportunity to express our gratitude to all those informative and supportive pillars who were directly or indirectly involved in the process of completion of my project.

It gives me immense pleasure to present this project report with gratitude to our internal guide **Prof. Arati Gavas**  without whose guidance this endeavor would not have been a success.

I also obliged **Principal Dr. Shyam Joshi.** and **Mr. S. J. Nalawade** (Incharge of Computer Science) for their whole hearted support. Last but not the least I thank all the people including the Laboratory and Library Staff who have directly or indirectly helped me.

I am thankful to other teaching and non teaching staff without whose help, this project would not be successful.

I am extremely delighted to mention about the valuable suggestions and encouragement from my Family and Friends also my Classmates and who have helped me from day one to make this project a success.

**Arati Baban Satape**

**Abstract**

Image Enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain feature of interest in an image. A familiar e.g. of enhancement is when we increase contrast of an image because “it looks better”. It is important to keep in mind that enhancement is a subjective area of image processing.

The rapid growth of digital imaging applications, including desktop publishing, multimedia, teleconferencing, and high-definition television (HDTV) has increased the need for effective and standardized image techniques.

We can do different enhancements of image by using this software like **T**ranslation, **S**caling, **R**otation, **B**lurring ,**S**harpening, **E**dge Detection, Mathematical Operations, Various Filters, Smooth, Sharpen, Encryption & Decryption etc.

**Project Overview**

**SOFTWARE NAME:**

**Image Enhancement**

**DEVELOPER:**

**Miss. Arati Baban Satape (MSc-II)**

**REQUIREMENTS:**

**Software:** jdk-6-windows-i586 (Java Software)

NetBeans IDE 6.0

**Hardware:** VDU having resolution at least 1024x768

**GUIDE:**

Prof. Miss Aarti Gavas.

**Objective & Scope of Project**

The main objective of the system is doing functions in Digital Image Processing.After digital images have been captured, and prior to initiating processing algorithm applications, each image should be evaluated with regard to its general characteristics, including noise, blur, background intensity variations, brightness and contrast, and the general pixel value distribution (histogram profile). Attention should be given to shadowed regions to determine how much detail is present, as well as bright features (or highlights) and areas of intermediate pixel intensity. This task is most easily accomplished by our software.

1. This application included various functions which operate on single platform.
2. User can easily make use of multiple functions together.
3. It is less time consuming to process image enhancement.
4. It’s a User friendly application.

**Applications of Image Processing**

* **Medical imaging**
* Locate tumors and other pathologies
* Measure tissue volumes
* Computer-guided surgery
* Diagnosis
* Treatment planning
* Study of anatomical structure
* **Locate objects in satellite images (roads, forests, etc.)**
* **Face recognition**
* **Fingerprint recognition**
* [**Traffic control systems**](http://en.wikipedia.org/w/index.php?title=Traffic_control_system&action=edit&redlink=1)
* [**Machine vision**](http://en.wikipedia.org/wiki/Machine_vision)
* **Photography Field**
* Colorize black/white photo
* Artistic Effects

**Hardware and Software Specification**

**Software requirements:**

* OS independent
* jdk-6-windows-i586 (Java Software)
* Net Beans IDE 6.0

**Hardware requirements:**

* + - 400 MHz processor
    - 256 Mb of RAM
    - 2 Mb of Hard Disk space
    - VDU having resolution at least 1024x768 pixels of screen resolution
    - 16 bit of color quality

**LITERATURE SURVEY**

****

**STATEMENT ABOUT THE PROBLEM**

So far we observed that most of the people who don’t aware about the beauties of Digital Image Processing because they think it is very hard subject to learn.

Optical microscopy is a rapidly developing field that has come to be highly dependent upon digital image processing techniques, both for aesthetic balance and cosmetic touches, as well as rehabilitation and analytical purposes. However, even when the microscope is configured correctly and performing optimally, captured digital images often display uneven backgrounds, excessive noise, aberration artifacts, poor contrast, out-of-focus regions, intensity fluctuations, and can also suffer from color shifts and color balance errors. In addition, images that appear perfectly sharp and crisp with excellent color saturation in the microscope can often be mangled by the image sensor to produce artifacts such as aliasing, camera noise, improper gamma correction, white balance shifts, poor contrast, and brightness fluctuations

We as a programmer thought to develop such software which will help aforementioned people in a way that they can solve their problems very quickly and accurate manner. We develop the software ”Image Enhancement”

**WHY PARTICULAR TOPIC CHOSEN?**

As I mentioned earlier that we wanted to provide the intermediate solutions for some Digital Image Processing problems which involves tedious and in most of the cases time consuming.

**About Image Enhancement**

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain feature of interest in an image. A familiar e.g of enhancement is when we increase contrast of an image because “it looks better”. It is important to keep in mind that enhancement is a very subjective area of image processing.

Image enhancement is the improvement of digital image quality (wanted e.g. for visual inspection or for machine analysis), without knowledge about the source of degradation.

The aim of image enhancement is to improve the interpretability or perception of information in images for human viewers, or to provide `better' input for other automated image processing techniques. Image enhancement techniques can be divided into two broad categories:

1. Spatial domain methods, which operate directly on pixels, and

2. frequency domain methods, which operate on the Fourier transform of an image.

Unfortunately, there is no general theory for determining what is `good' image enhancement when it comes to human perception. If it looks good, it is good! However, when image enhancement techniques are used as pre-processing tools for other image processing techniques, then quantitative measures can determine which techniques are most appropriate.

That’s why we thought to exploit this efficiency of computers to solve several Digital Image Processing problems using such machine. And we are also well understood the features of Java as a programming language so, we developed this software

**Feasibility Study**

The initial investigation points to the question whether the project is feasible. A feasibility is conducted to identify the best system that meets the all the requirements. This includes an identification description, an evaluation of the proposed systems and selection of the best system for the job the requirements of the system are specified with a set of constraints such as system objectives and the description of the out puts. It is then duty of the analyst to evaluate the feasibility of the proposed system to generate the above results. Three key factors are to be considered during the feasibility study.

**Operation Feasibility:** An estimate should be made to determine how much effort and care will go into the developing of the system including the training to be given to the user. Usually, people are reluctant to changes that come in their progression. The computer initialization will certainly affected the turn over, transfer and employee job status. Hence an additional effort is to be made to train and educate the users on the new way of the system.

**Technical Feasibility:** The main consideration is to be given to the study of available resources of the organization where the software is to be implemented. Here the system analyst evaluates the technical merits of the system giving emphasis on the performance, Reliability, maintainability and productivity.  
By taking the consideration before developing the proposed system, the resources availability of the organization was studied. The organization was immense computer facilities equipped with sophisticated machines and the software hence this technically feasible.

**Economic Feasibility:**  
 Economic feasibility is the most important and frequently used method for evaluating the effectiveness of the proposed system. It is very essential because the main goal of the proposed system is to have economically better result along with increased efficiency. Cost benefit analysis is usually performed for this purpose. It is the comparative study of the cost verses the benefit and savings that are expected from the proposed system. Since the organization is well equipped with the required hard ware, the project was found to be economically.

**Introduction**

**What is an image?**

An image may be defined as a two-dimensional function, f(x, y), where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates(x, y) is called the intensity or gray level of the image at that point. When x, y, and the amplitude values of f are all finite, discrete quantities we call image a digital image. Digital image composed of a finite number of elements, each of which has a particular location and value. These elements are referred as to picture element , image elements, pels, pixels. Pixel is used to denote the element of digital image.[8]

A digital image is a representation of a two-dimensional image using ones and zeros (binary). Depending on whether or not the image resolution is fixed, it may be of vector or raster type. Without qualifications, the term "digital image" usually refers to raster images also called bitmap images.

An analog image converted to numerical form so that it can be stored and used in a computer. The image is divided into a matrix of small regions called picture elements or pixels. At sub-satellite point each pixel represents a specific amount of area.[8]

**Why Digital Image Processing Required?**

Digital image processing plays a vital role in the analysis and interpretation of remotely sensed data. Especially data obtained from Satellite Remote Sensing, which is in the digital form, can best be utilized with the help of digital image processing. Image enhancement and information extraction are two important components of digital image processing. Image enhancement techniques help in improving the visibility of any portion or feature of the image suppressing the information in other portions or features. Information extraction techniques help in obtaining the statistical information about any particular feature or portion of the image.Interest in digital image processing methods system s from two principal application areas:

* Improvement of pictorial information for human interpretation and
* Processing of image data for storage, transmission, and representation for autonomous machine perception.

Digital image processing enables the enhancement of visibility for detail in images using algorithms that apply arithmetic and statistical procedures to stored pixel values, instead of the classical darkroom manipulations for filtration of time-dependent voltages necessary for analog images and video signals. Even though many image processing algorithms are extremely powerful, the average user often applies operations to digital images without concern for the underlying principles behind these manipulations. The images that result from careless manipulation are often severely degraded or otherwise compromised with respect to those that could be produced if the power and versatility of the digital processing software were correctly utilized.[8]

**Three type of Computerized Processes used for Image Processing**

**Low-Level Processes:**

Low-level processes involve primitive operations such as image processing to reduce noise, contrast enhancement and image sharpening. Low- level processes is characterized by the fact that both its input and output are images.

**Mid-Level Processes:**

Mid-level processing on images involves tasks such as segmentation, description of those objects to reduce them to a form suitable for computer processing and classification of individual objects. Mid- level processes is characterized by the fact that its input generally image and output are attributes extracted from those image.

**Higher-Level Processes:**

Higher-level processing involves “making sense” of an ensemble of recognized objects, as in image analysis and at far end of the continuum, performing the cognitive functions normally associated with vision. [8]

[**Overview of Image Processing and Analysis**](http://micro.magnet.fsu.edu/primer/digitalimaging/russ/overview.html)

***Image Processing*** operates on images and results in images, with changes intended to improve the visibility of features, or to make the images better for printing or transmission, or to facilitate subsequent analysis. ***Image Analysis*** is the process of obtaining numerical data from images. This is usually accomplished by a combination of measurement and processing operations. The data may subsequently be analyzed statistically, or used to generate graphs or other visualizations.

[**Correcting Image Defects**](http://micro.magnet.fsu.edu/primer/digitalimaging/russ/correcting.html)

Most images include some imperfections, the result of the inherent limitations in illumination, optics, camera or the specimen itself. Many of these can be improved by processing. If color images are being acquired, it is first of all important to understand something about color representations. Then the removal of noise may be required, using methods that depend on the noise source. After correction for non-uniform illumination, the contrast and brightness can be adjusted for optimum visibility of detail. Finally, limitations in image focus may be addressed.

**Pre-Processing Evaluation of Digital Images**

After digital images have been captured, and prior to initiating processing algorithm applications, each image should be evaluated with regard to its general characteristics, including noise, blur, background intensity variations, brightness and contrast, and the general pixel value distribution (histogram profile). Attention should be given to shadowed regions to determine how much detail is present, as well as bright features (or highlights) and areas of intermediate pixel intensity. This task is most easily accomplished by importing the image into one of the popular software editing programs, such as Adobe Photoshop, Corel Photo-Paint, Macromedia Fireworks, or Paint Shop Pro.

Image processing is the field of signal processing where both the input and output signals are images. Images can be thought of as two-dimensional signals via a matrix representation, and image processing can be understood as applying standard one-dimensional signal processing techniques to two-dimensional signals. Image processing is a very important subject, and finds applications in such fields as photography, satellite imaging, medical imaging, and image compression, just to name a few.

**Working Of Function**

Image Enhancementprovides different types of operation on an image, are as follows:

[**Geometric Transformation: Transform Rotate, Flip, Scale, and Zoom**](http://micro.magnet.fsu.edu/primer/java/digitalimaging/processing/panscrollzoom/index.html)

The microscopic must often arrange digital images on the output display device for purposes of visually comparing image details, or for illustrating differences between two or more images. Such display operations are facilitated through software packages that enable interactive panning, scrolling, flipping, scaling, and zooming of digital images. Zooming to enlarge image details is useful for the visualization of small structures present in a digital image, and scaling is often necessary to format a digital image to fit within the boundaries of a display medium, as in the case of displaying a collection of thumbnail images.[1]

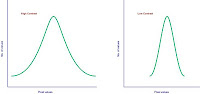
**Brightness:**

Brightness is subjective descriptor of light perception that is practically impossible to measure. It embodies the achromatic notion of intensity and is one of the key factors in describing color sensation. It is one of the most significant pixel characteristics. It is involved in many image-editing algorithms such as contrast or shadow/highlight.

The **brightness** (or **luminous brightness**) of a digital image is a measure of relative intensity values across the pixel array after the image has been acquired with a digital camera or digitized by an analog-to-digital converter. Brightness should not be confused with intensity (more accurately termed radiantintensity), which refers to the magnitude or quantity of light energy actually reflected from or transmitted through the object being imaged by an analog or digital device. Instead, in terms of digital image processing, brightness is more properly described as the measured intensity of all the pixels comprising an ensemble that constitutes the digital image after it has been captured, digitized, and displayed. Pixel brightness is an important factor in digital images, because (other than color) it is the only variable that can be utilized by processing techniques to quantitatively adjust the image.[8]

**Contrast:**

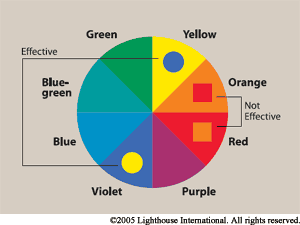
The contrast adjustment is a multiplication of the pixel values with a constant. The contrast of an image is defined by the distribution of the color values of the pixel of the image i.e in a high contrast image, the color values of pixels exist over wide distribution. On the other hand, a low contrast image, they exist in narrow distribution. Hence color values are grouped together tightly over short range of value for low contrast and are grouped loosely over a wide range of values for high contrast images.

****

**Color Contrast**

Contrast is the difference in luminance and/or color that makes an object (or its representation in an image or display) distinguishable. In visual perception of the real world, contrast is determined by the difference in the color and brightness of the object and other objects within the same field of view.[5]

The color contrast of web page elements depends on the differences between their brightness levels and their hues. If the brightness and hue differences between two colors are above a certain threshold, the colors have sufficient contrast. We can calculate color contrast if you know the RGB (red-green-blue) values of two colors. Designing web pages with enough contrast between the background and text improves user experience.



**Size**

In computer graphics, **image scaling** is the process of resizing a digital image. Scaling is a non-trivial process that involves a trade-off between efficiency, smoothness and sharpness. As the size of an image is increased, so the pixels which comprise the image become increasingly visible, making the image appear "soft". Conversely, reducing an image will tend to enhance its smoothness and apparent sharpness.[5]

Two standard scaling algorithms are bilinear and bicubic interpolation. Filters like these work by interpolating pixel color values, introducing a continuous transition into the output even where the original material has discrete transitions. Nearest-neighbor interpolation preserves these sharp edges, but it increases  Several approaches have been developed that attempt to optimize for bitmap art by interpolating areas of continuous tone, preserve the sharpness of horizontal and vertical lines and smooth all other curves.

**Aspect Ratio**

The horizontal-to-vertical dimensional ratio of a digital image is referred to as the **aspect ratio** of the image and can be calculated by dividing the horizontal width by the vertical height. The recommended **NTSC** (National Television Systems Committee) commercial broadcast standard aspect ratio for television and video equipment is 1.33, which translates to a ratio of 4:3, where the horizontal dimension of the image is 1.33 times wider than the vertical dimension. In contrast, an image with an aspect ratio of 1:1 (often utilized in closed circuit television or **CCTV**) is perfectly square. By adhering to a standard aspect ratio for display of digital images, gross distortion of the image, such as a circle appearing as an ellipse, is avoided when the images are displayed on remote platforms.[5]



The 4:3 aspect ratio standard, widely utilized for television and computer monitors, produces a display that is four units wide by three units high. For example, a 32-inch television (measured diagonally from the lower left-hand corner to the upper right-hand corner) is 25.6 inches wide by 19.2 inches tall. The standard aspect ratio for digital high-definition television (**HDTV**) is 16:9 (or 1.78:1), which results in a more rectangular screen. Sometimes referred to as **widescreen** format (see Figure 2), the 16:9 aspect ratio is a compromise between the standard broadcast format and that commonly utilized for motion pictures. This ratio has been determined to provide the best compromise at eliminating or decreasing the size of black bars for **letterbox** format movies, while minimizing the size of bars required to fit traditional 4:3 broadcasts into screens using the wider format.

**Black and White**

In photography and computing, a **grayscale** or **greyscale** digital image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information. Images of this sort, also known as black-and-white, are composed exclusively of shades of gray, varying from black at the weakest intensity to white at the strongest.[1]

Grayscale images are distinct from one-bit bi-tonal black-and-white images, which in the context of computer imaging are images with only the two  color black and white (also called *bi-level* or *binary images*). Grayscale images have many shades of gray in between. No matter what pixel depth is used, the binary representations assume that 0 is black and the maximum value (255 at 8 bpp, 65,535 at 16 bpp, etc.) is white.

**Rotate**

The rotation operator performs a geometric transform which maps the position (x1,y1) of a picture element in an input image onto a position (x2,y2) in an output image by rotating it through a user-specified angle Ø about an origin O. In most implementations, output locations (x2,y2) which are outside the boundary of the image are ignored. Rotation is most commonly used to improve the visual appearance of an image, although it can be useful as a preprocessor in applications where directional operators are involved. Rotation is a special case of affine transformation.

**Mathematical and Logical Operations**

**NOT**

Logical NOT or invert is an operator which takes a binary or graylevel image as input and produces its photographic negative, i.e. dark areas in the input image become light and light areas become dark.

Each pixel in the input image having a logical 1 (often referred to as foreground) has a logical 0 (associated with the background in the output image and vice versa. Hence, applying logical NOT to a binary image changes its polarity.[6]

**Log**

General form of log transformation is

S = c log (1+r) where c is a constant and r>=0.

This transformation maps a narrow range of low gray-level values in input image into wider range of output levels. The opposite is true of higher values of input levels. We use transformation of this type to expand the values of dark pixel in an image while compressing the higher level values.[8]

**Filters**

**Gaussian Blur**

A **Gaussian blur** (also known as Gaussian smoothing) is the result of blurring an image by a Gaussian function. It is a widely used effect in graphics software, typically to reduce image noise and reduce detail. The visual effect of this blurring technique is a smooth blur resembling that of viewing the image through a translucent screen, distinctly different from the bokeh effect produced by an out-of-focus lens or the shadow of an object under usual illumination. Gaussian smoothing is also used as a pre-processing stage in computer vision algorithms in order to enhance image structures at different scales—see scale-space representation and scale-space implementation.[1]

Mathematically, applying a Gaussian blur to an image is the same as convolving the image with a Gaussian function; Since the Fourier transform of a Gaussian is another Gaussian, applying a Gaussian blur has the effect of reducing the image's high-frequency components; a Gaussian blur is thus a low pass filter.

**Median**

In signal processing, it is often desirable to be able to perform some kind of noise reduction on an image or signal. The **median filter** is a nonlinear digital filtering technique, often used to remove noise. Such noise reduction is a typical pre-processing step to improve the results of later processing (for example, edge detection on an image). Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise .The main idea of the median filter is to run through the signal entry by entry, replacing each entry with the median of neighboring entries.[8]

**Mean**

**Mean filter**, or **Average filter** is windowed filter of linear class, that smoothes signal (image). The filter works as low-pass one. The basic idea behind filter is for any element of the signal (image) take an average across its neighborhood. To understand how that is made in practice, let us start with window idea. The Average (mean) filter smooth image data, thus eliminating noise. This filter performs spatial filtering on each individual pixel in an image using the grey level values in a square or rectangular window surrounding each pixel.[1]

The arithmetic mean is the "standard" average, often simply called the "mean".

 \bar{x} = \frac{1}{n}\cdot \sum_{i=1}^n{x_i} 

**Min-Max**

Maximum and minimum filters attribute to each pixel in an image a new value equal to the maximum or minimum value in a neighborhood around that pixel. The neighborhood stands for the shape of the filter. Maximum and minimum filters have been used in contrast enhancement and normalization, texture description, edge detection, and thresholding .The filters are grey value analogues of dilation and erosion.[8]

## Smoothing

## 

The aim of image smoothing is to diminish the effects of camera noise, spurious pixel values, missing pixel values etc. There are many different techniques for image smoothing; we will consider neighborhood averaging and edge-preserving smoothing.

### Neighbourhood Averaging

Each point in the smoothed image, F(x,y)is obtained from the average pixel value in a neighborhood of (*x*,*y*) in the input image.

For example, if we use a 3×3neighborhood around each pixel we would use the mask

|  |  |  |
| --- | --- | --- |
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |

Each pixel value is multiplied by 1/9, summed, and then the result placed in the output image. This mask is successively moved across the image until every pixel has been covered. That is, the image is convolved with this smoothing mask (also known as a spatial filter or kernel).

However, one usually expects the value of a pixel to be more closely related to the values of pixels close to it than to those further away. This is because most points in an image are spatially coherent with their neighbours; indeed it is generally only at edge or feature points where this hypothesis is not valid. Accordingly it is usual to weight the pixels near the centre of the mask more strongly than those at the edge.

Some common weighting functions include the rectangular weighting function above (which just takes the average over the window), a triangular weighting function, or a Gaussian. In practice one doesn't notice much difference between different weighting functions, although Gaussian smoothing is the most commonly used. Gaussian smoothing has the attribute that the frequency components of the image are modified in a smooth manner. Smoothing reduces or attenuates the higher frequencies in the image. [8]

**Thresholding**

Thresholding makes color changes across a programmer-determined "boundary," or threshold, more obvious (similar to how the contour lines on a map make altitude boundaries more obvious). This technique uses a specified threshold value, minimum value, and maximum value to control the color component values for each pixel of an image. Color values below the threshold are assigned the minimum value. Values above the threshold are assigned the maximum value.

The thresholding process is performed for each color component of each pixel. When the operation is complete, the color components of the destination image pixels will contain either the minimum value or the maximum value. For example, consider what happens to an image when a thresholding operation is performed with a minimum of 0 and a maximum of 255. After the image is processed, the red, green, and blue values of the pixels will be either 0 or 255.

**Emboss**

An image appears to be embossed when highlights and shadows replace light/dark boundaries and low contrast areas are set to a gray background. **Embossing** is achieved by setting pixels black except where there is a change in intensity.

|  |  |  |
| --- | --- | --- |
| Description: unembossed Original | Description: true emboss True emboss | Description: quick emboss Quick emboss |

Here are two methods for creating an embossed image. The **true emboss** calculates the first derivative of the image and combines it with a uniform grayscale. The **quick emboss** creates an emboss-like effect by combining the image with its negative at a slight offset.[1]

**Sharpen**

The main aim in image sharpening is to highlight fine detail in the image, or to enhance detail that has been blurred (perhaps due to noise or other effects, such as motion). With image sharpening, we want to enhance the high-frequency components; this implies a spatial filter shape that has a high positive component at the centre.

**Contrast Stretching**

Contrast stretching (often called normalization) is a simple image enhancement technique that attempts to improve the contrast in an image by `stretching' the range of intensity values it contains to span a desired range of values, e.g. the the full range of pixel values that the image type concerned allows. It differs from the more sophisticated histogram equalization in that it can only apply a linear scaling function to the image pixel values. As a result the `enhancement' is less harsh. (Most implementations accept a gray level image as input and produce another graylevel image as output.) [3]

**Equalize Histogram**

Histogram equalization is a common technique for enhancing the appearance of images. Suppose we have an image which is predominantly dark. Then its histogram would be skewed towards the lower end of the grey scale and all the image detail is compressed into the dark end of the histogram. If we could `stretch out' the grey levels at the dark end to produce a more uniformly distributed histogram then the image would become much clearer. Histogram equalization involves finding a grey scale transformation function that creates an output image with a uniform histogram (or nearly so).

|  |
| --- |
| **Figure 2:** The original image and its histogram, and the equalized versions. Both images are quantized to 64 grey levels. |
| \begin{figure} \par \centerline{ \psfig {figure=figure52.ps,width=12cm} } \par\end{figure} |

**Binary**

Binary images are images that have been quantised to two values, usually denoted 0 and 1, but often with pixel values 0 and 255, representing black and white.

Binary images are used in many applications since they are the simplest to process, but they are such an impoverished representation of the image information that their use is not always possible. However, they are useful where all the information you need can be provided by the silhouette of the object and when you can obtain the silhouette of that object easily.

**Edge Detection**

   An edge in an image is a boundary or contour at which a significant change occurs in some physical aspect of an image, such as the surface reflectance, illumination, or the distances of the visible surfaces from the viewer. Changes in physical aspects manifest themselves in a variety of ways, including changes in intensity, color, and texture. In our discussion, we are concerned only with the changes in image intensity.

Detecting edges is very useful in a number of contexts. For example, in a typical image understanding task such as object identification, an essential step is to segment an image into different regions corresponding to different objects in the scene. Edge detection is often the first step in image segmentation. As another example, one approach to the development of a low bit-rate image coding system is to code only the detected edges. it is well known that an image that consists of only edges is highly intelligible.

**Encryption and Decryption**

A method of image encryption and decryption is proposed for optical security systems. A phase-coded image to be encrypted together with a random phase pattern is Fourier transformed and the result of the complex-valued data is used as an encrypted pattern. The decryption is simply performed by an inverse-Fourier transform for the addition of the encrypted pattern and the Fourier transform of the random phase. The intensity of the inverse-Fourier transformed image gives the exact result for the decryption. Further, the binarization of the encrypted pattern is performed for the easiness of the optical and electronic readout of the images, and it also gives rise to the enhancement of the degree of security for the encryption. The binary pattern is optimized by a statistical iteration technique and an excellent decryption image is obtained by the optimization.

Basically Image Encryption means that, convert the image into unreadable format. Internet based communications are evolving at a tremendous rate. Encryption of data has become an important way to protect data resources especially on the Internet, intranets and extranets. Encryption involves applying special mathematical algorithms and keys to transform digital data into cipher code before they are transmitted and decryption involves the application of mathematical algorithms and keys to get back the original data from cipher code. The goal of security management is to provide authentication of users, and integrity, accuracy and safety of data resources. Moreover, an image-based data requires more effort during encryption and decryption. The model for encryption and decryption of an image using suitable user-defined key is developed with the sameobjective.

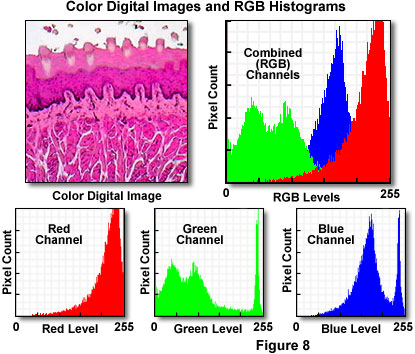
**Histogram**

**Histogram**

A Histogram of a digital image with gray level in the range [0 , L-1] is a discrete function (hr)=nk , where rk is the gray level & nk is the number of pixel in the image having gray level rk.

**Color Histogram**

Histograms of color digital images are a composite of three grayscale histograms that are computed and displayed for each color component (usually red, green, and blue). Color histograms can represent RGB color-space, HSI models, or any other color space model necessary for digital image processing algorithms. These histograms can be displayed simultaneously in a superimposed fashion, or segregated into individual graphs to help determine brightness distributions, contrast, and dynamic ranges of the individual color components.[5]



Presented in Figure 8 is a typical full color digital image captured with an optical microscope. The specimen is a thin section of mammalian taste buds stained with eosin and hematoxylin and imaged under brightfield illumination mode. Appearing to the right of the digital image is the RGB histogram, which contains superimposed pixel distributions for the three (red, green, and blue) color channels. Beneath the digital image and the RGB histogram are individual histograms representing the red, green, and blue channels, respectively. Note that the distribution of intensity levels is highest in the red channel, which corresponds to the rather pronounced dominance of reddish tones in the digital image. The bimodal green channel indicates a large degree of contrast in this color channel, while the blue channel presents a histogram having a relatively well-distributed intensity range.

In image processing and photography, a **color histogram** is a representation of the distribution of colors in an image. For digital images, a color histogram represents the number of pixels that have colors in each of a fixed list of color ranges, that span the image's color space, the set of all possible colors.

**The color histogram** can be built for any kind of color space, although the term is more often used for three-dimensional spaces like RGB or HSV. For monochromatic images, the term **intensity histogram** may be used instead. For multi-spectral images, where each pixel is represented by an arbitrary number of measurements (for example, beyond the three measurements in RGB), the color histogram is *N*-dimensional, with N being the number of measurements taken. Each measurement has its own wavelength range of the light spectrum, some of which may be outside the visible spectrum.

If the set of possible color values is sufficiently small, each of those colors may be placed on a range by itself; then the histogram is merely the count of pixels that have each possible color. Most often, the space is divided into an appropriate number of ranges, often arranged as a regular grid, each containing many similar color values. The color histogram may also be represented and displayed as a smooth function defined over the color space that approximates the pixel counts.

Color histograms are flexible constructs that can be built from images in various color spaces, whether RGB, rg chromaticity or any other color space of any dimension. A histogram of an image is produced first by discretization of the colors in the image into a number of bins, and counting the number of image pixels in each bin. For example, a Red–Blue chromaticity histogram can be formed by first normalizing color pixel values by dividing RGB values by R+G+B, then quantizing the normalized R and B coordinates into N bins each. A two-dimensional histogram of Red-Blue chromaticity divide in to four bins (*N*=4) might yield a histogram that looks like this table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | red | | | |
| 0-63 | 64-127 | 128-191 | 192-255 |
| blue | 0-63 | 43 | 78 | 18 | 0 |
| 64-127 | 45 | 67 | 33 | 2 |
| 128-191 | 127 | 58 | 25 | 8 |
| 192-255 | 140 | 47 | 47 | 13 |

A histogram can be N-dimensional. Although harder to display, a three-dimensional color histogram for the above example could be thought of as four separate Red-Blue histograms, where each of the four histograms contains the Red-Blue values for a bin of green (0-63, 64-127, 128-191, and 192-255).

## 

**Class Diagramm**

|  |
| --- |
| Enhancer |
| Int maximageframesize;  int width=100;  int height=100;  int widthbuffer=0;  int heightbuffer=0;  int[][][] colouray;  int rgbs[];  int rgbsbuffer[];  int rgbsundoarray[];  int widthundo=0;  int heightundo=0;  int regions=5;  int encryptationcycles=0;  boolean imageopen=false;  boolean dialogopen=false;  boolean otherframeopen=false;  Color colorsel1;  Color colorsel2; |
| Loadimage()  Saveimage()  invert()  blackandwhite()  smoothing()  filtermedian()  findedges()  squareroot()  log()  square()  reciprocal()  flipvertical()  fliparbitrary()  contrast()  distribute()  colour()  showinfo()  binary()  bright()  adjustsize()  pencilsketch()  decrypt()  swapcolor()  maximumfilter()  minimumfilter()  booleannot()  findedgessuperb()  emboss()  anycolorswap()  contraststretching()  histogramequalisation()  gammacorrection()  function() |

**Analysis And Design**

****

**JAVA AS A PROGRAMMING LANGUAGE**

**C:\Program Files\Java\jdk1.6.0\demo\plugin\jfc\Java2D\src\images\java_logo.png**

**“Java is a simple, object-oriented, distributed, interpreted, robust, secure, architecture neutral, portable, multithreaded, and dynamic."**

**Java is simple**

Java is considered much simpler and easy to use object-oriented programming language when compared to the other popular programming language. Java has replaced the complexity of multiple inheritance in C++ with a simple structure called interface, and also has eliminated the use of pointers. Java uses automatic memory allocation and garbage collection where else C++ requires the programmer to allocate memory and to collect garbage. The clean syntax makes Java programs easy to write and read.

**Java is Object-oriented**

Java is purely object oriented language. Object-oriented programming models the real world. Everything in the world can be modeled as an object.

**Java is Distributed**

Distributed computing involves several computers on a network working together. Java is designed to make distributed computing easy with the networking capability that is inherently integrated into it.

**Portability: Program once, Run anywhere (Platform Independence)**

Java is platform independent. Java runs on most major hardware and software platforms, including Windows 95 and NT, the Macintosh, and several varieties of UNIX. JAVA programs become more portable. Any hardware and operating system dependencies are removed. Moving between OS platforms requires recompilation, as a minimum and significant redesign, in most cases.

**Java is Interpreted**

An interpreter is needed in order to run Java programs. The programs are compiled into Java Virtual Machine code called byte code. The byte code is machine independent and is able to run on any machine that has a Java interpreter. With Java, the program need only be compiled once, and the byte code generated by the Java compiler can run on any platform.

**Security**

Java is one of the first programming languages to consider security as part of its design. The Java language, compiler, interpreter, and runtime environment were each developed with security in mind.

**Reliability**

Security and reliability go hand in hand. Java provides multiple levels of reliability measures, beginning with the Java language itself. The Java compiler provides several levels of additional checks to identify type mismatches and other inconsistencies. The Java runtime system duplicates many of the checks performed by the compiler and performs additional checks to verify that the executable byte codes form a valid Java program.

**Multimedia: Images, Sounds and Animation**

The sizzle of JAVA is MULTIMEDIA - Sounds, Images, Graphics and Video. In this growing age of multimedia, new computers are known as "multimedia ready" with CD-Rom drives, sound cards, 3D accelerator cards and other new special sound or graphic technology capabilities. We also need programming languages that make creating multimedia easy. Most programming languages do not have built-in multimedia capabilities. JAVA, however through the packages of classes that are an integral part of the Java programming world, provides extensive multimedia facilities that will enable a programmer to start developing powerful multimedia applications immediately.

**The Virtual Machine: Java VM**

This VM sits, metaphorically, between the Java program and the machine it is running on, offering the program an "abstract computer" that executes the Java code and guarantees certain behaviors regardless of the underlying hardware or software platform. Java compilers thus turn Java programs not into assembly language for a particular machine but into a platform-neutral "byte code" that the machine-specific VM interprets on the fly.

**Java is Robust**

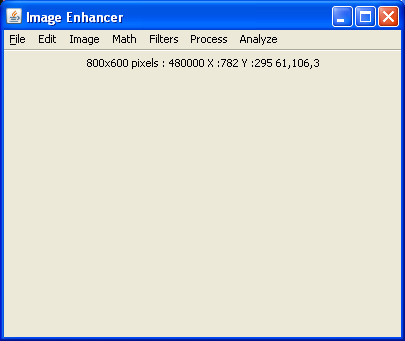
Robust means reliable and no programming language can really assure reliability. Java puts a lot of emphasis on early checking for possible errors. Java does not support pointers, which eliminates the possibility of overwriting memory and corrupting data. Java has a runtime exception-handling feature to provide programming support for robustness, and can catch and respond to an exceptional situation so that the program can continue its normal execution and terminate gracefully when a runtime error occurs.

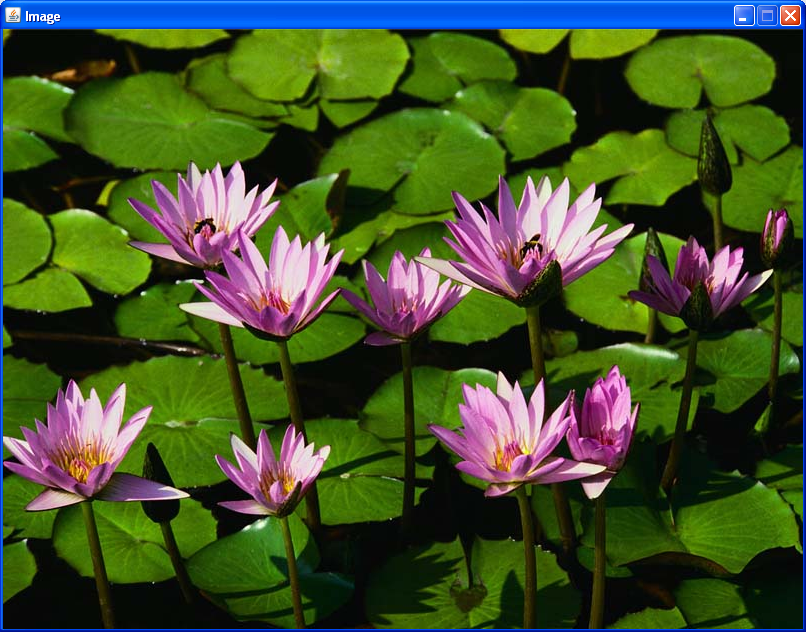
**Java is Multithreaded**

Multithreaded is the capability for a program to perform several tasks simultaneously within a program. For instance, playing an mp3 file while downloading the file would be considered multithreading. In Java, multithreaded programming has been smoothly integrated into it. Multithreading is especially useful in graphical user interface (GUI) and network programming. In GUI programming, many things can occur at the same time. In network programming, a server can serve multiple clients at the same time. Multithreading is a necessity in visual and network programming.

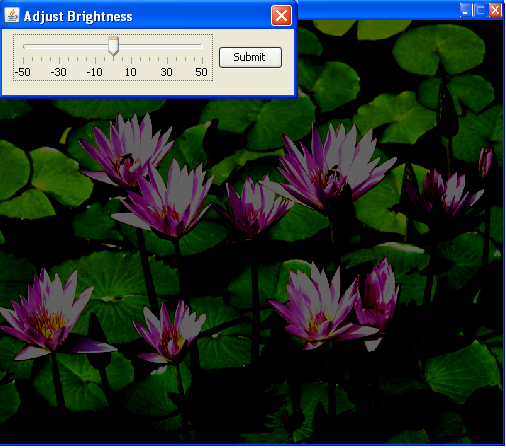
**Screen Layouts**

**Home Screen**





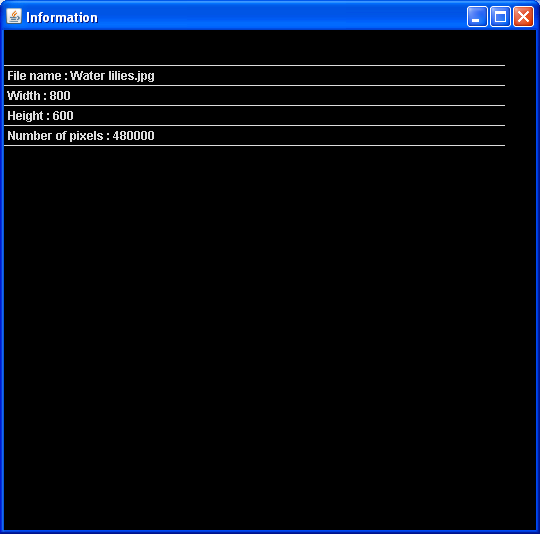
**Adjust Brightness**



**Black and white**



**Showinfo**

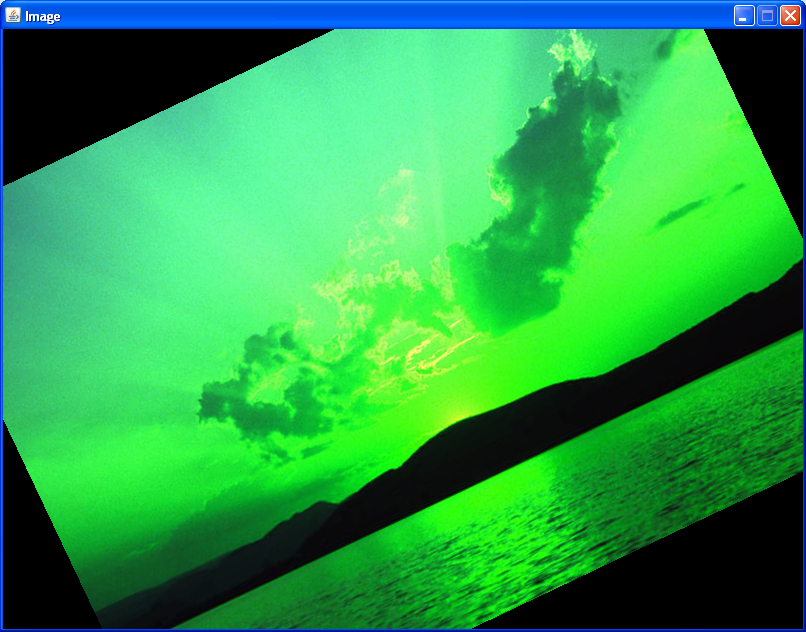


**Rotate-FlipArbitrary**

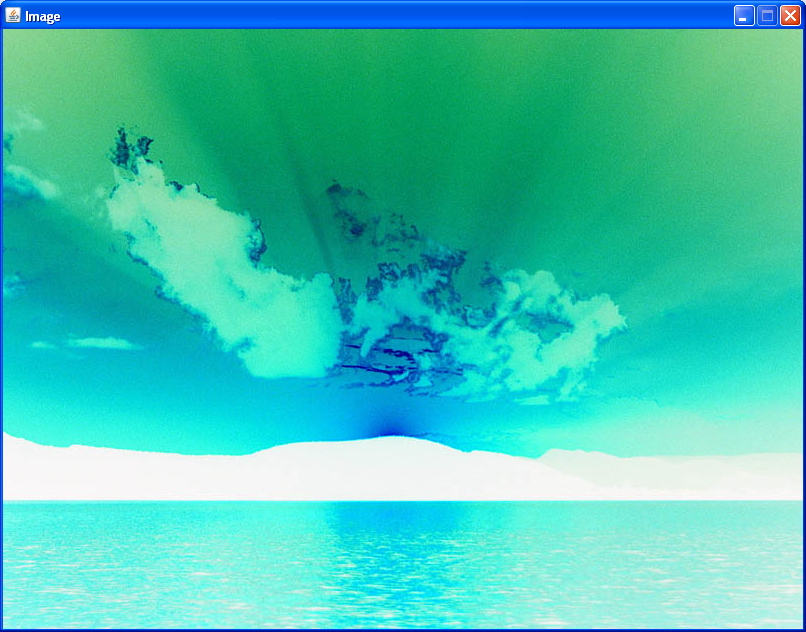


**Swap any color**

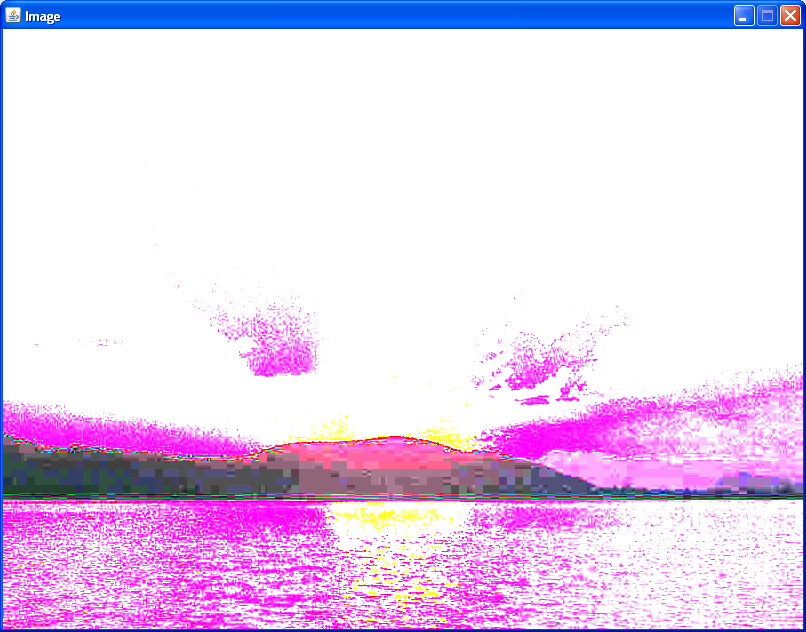
****



**Math-Not**



**Math-Square**



**Image**



**Smooth Image**

****

**Sharpen Image**



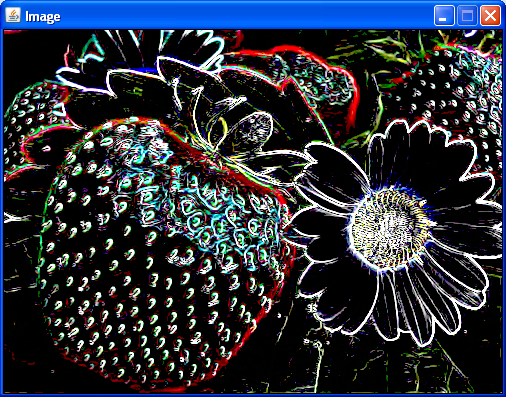
**Binary**



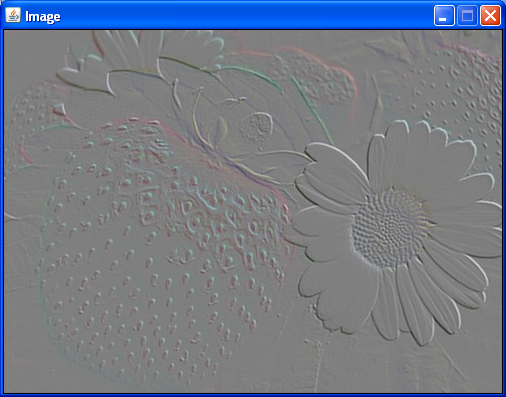
**Image**



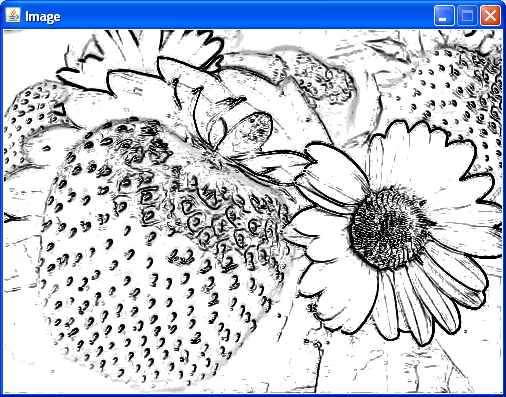
**Find Edges**



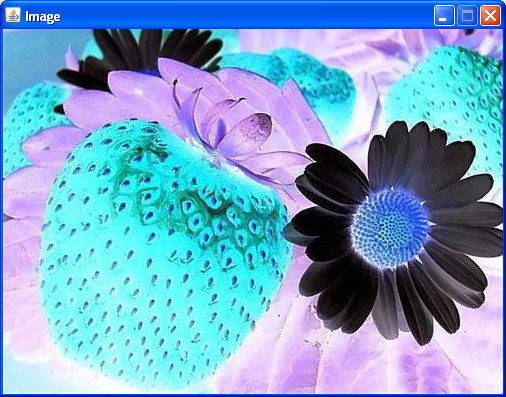
**Emboss**



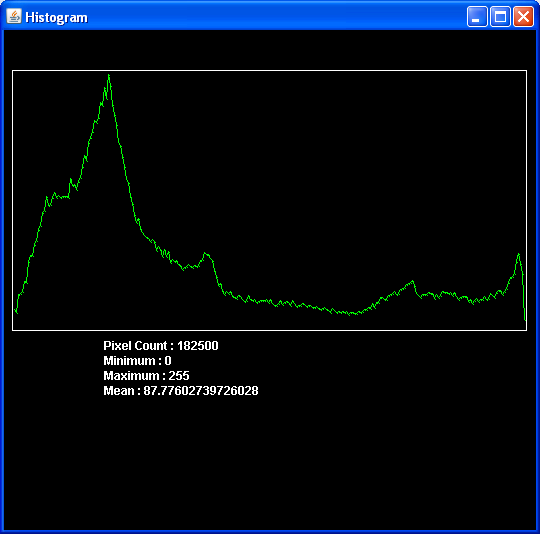
**Pencil Sketch**



**Invert**



**Histogram**



**Limitations**

This software is designed to accomplish the requisite of MSc-II Computer Science ‘Project Work’ as a ‘Co-curricular Activity’. I think have paid dedicated attention towards the development of this software. But there is lot of scope to improve.

Another thing is to mention here is that we are going to put it against its intended users. Probably they may find some drawbacks in this software. They all are welcome to put their queries and suggestions on my E-Mail ID: [aratisatape@gmail.com](mailto:aratisatape@gmail.com). We promise that we will try hard to overcome all your queries and needs as mentioned in ‘Future Enhancement’.

**Future Enhancement**

Our Future Enhancement is coloring the gray-scale images into the color images with referenced color image.The general problem of adding chromatic values to a gray-scale image has no exact solution, the current approach attempts to provide a method to minimize the amount of human labor required for this task, rather than choosing RGB colors to individual components, we convert the entire color to the gray-scale image by matching luminance and between the images.

Reference Target Final

Image Image Image

**Conclusions**

The power of digital image processing to extract information from noisy or low-contrast images and to enhance the appearance of these images has led some investigators to rely on the technology instead of optimally adjusting and using the microscope or image sensor. Invariably, beginning with a higher-quality optical image, free of dirt, debris, noise, aberration, glare, scratches, and artifacts, yields a superior electronic image. Careful adjustment and proper calibration of the image sensor will lead to a higher-quality digital image that fully utilizes the dynamic range of both the sensor and the digital image processing system.

**References**

**Websites**

[1] http://en.wikipedia.org/wiki/Image\_processing

[2] http://en.wikipedia.org/wiki/Histogram

[3]http://jhlabs.com/image processing

[4] http://en.wikipedia.org/wiki/contraststreching

[5] http://micro.magnet.fsu.edu/primer/digitalimaging/javaindex.html

[6] http://homepages.inf.ed.ac.uk/rbf/HIPR2/arthops.htm

[7] http://www.google.com

**Text Books**

[8] R.C.Gonsales R.E.Woods, “Digital Image Processing”.

[9] Anil K. Jain, “Fundamentals of Image Processing”, PHI.