Image Processing using Scilab

Project By

Prashant Santosh Badgujar
Kalpesh Anil Wani

Under the Guidance of

Mrs. Nayana C. Borase

शासकीय तंत्रनिकेतन, जळगाव

Department Of Computer Engineering
Government Polytechnic, Jalgaon

Semester – VI, Third Year
2019 – 2020

Department of Computer Engineering
Government Polytechnic, Jalgaon

शासकीय तंत्रनिकेतन, जळगाव

CERTIFICATE

This is to certify that

1. **Prashant Santosh Badgujar**
2. **Kalpesh Anil Wani**

of CO6I (2019-20) have successfully completed Project on **“Image Processing using Scilab”,** under the guidance of **“Mrs. Nayana C. Borase mam”** in partial fulfilment of the requirement for the award of diploma in computer engineering from Maharashtra State Board of Technical Education, Mumbai at Government Polytechnic, Jalgaon.

Mrs. Nayana C. Borase	**Dr. M.V. Ingale**	**Dr. P.P.Chaudhari**
GUIDE	**PRINCIPLE**	**HOD**

Department of Computer Engineering
Government Polytechnic, Jalgaon

CERTIFICATE

This is to certify that

1. **Prashant Santosh Badgujar**
2. **Kalpesh Anil Wani**

of CO6I (2019-20) have successfully completed Project on **"Image Processing using Scilab",** under the guidance of **"Mrs. Nayana C. Borase mam"** in partial fulfilment of the requirement for the award of diploma in computer engineering from Maharashtra State Board of Technical Education, Mumbai at Government Polytechnic, Jalgaon.

Mrs. Nayana Borase

GUIDE **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We convey our warm gratitude and sincere thanks to Mrs. Nayana Borase Mam for her Valuable guidance, constant encouragement and valuable suggestions, without which the Present capstone project would not have come to its shape.
We also like to extend our gratitude to the principal Sir “Mr M.V. Ingale” and Head of Computer department “Mr P.P. Chaudhari” Sir for providing us with all the facility that was required. We are equally grateful to all present faculty members of computer department including Teaching and non-teaching staff for providing require academic facilities in accomplishing our project work. We express our gratitude to the Author, editors, and publishers of the book which helped us in completing our projects.
Many people, especially our team members itself, have made valuable comment, suggestion on this project which gave us inspiration to improve our project.

We would also thank to all of them who helped us to complete this project.

1. Prashant Badgujar

2. Kalpesh Wani

(CO6I)

Table of Contends

Table of Diagrams

Chapter 1: Introduction

1.1 Overview:

Scilab is a programming language associated with a rich collection of numerical algorithms covering many aspects of scientific computing problems. From the software point of view, Scilab is an interpreted language. This generally allows to get faster development processes, because the user directly accesses to a high level language, with a rich set of features provided by the library. The Scilab language is meant to be extended so that user-defined data types can be defined with possibly overloaded operations. Scilab users can develop their own module so that they can solve their particular problems. The Scilab language allows to dynamically compile and link other languages such as FORTRAN and C: this way, external libraries can be used as if they were a part of Scilab built-in features. Scilab also interfaces Lab VIEW, a platform and development environment for a visual programming language from National Instruments. From the license point of view, Scilab is a free software in the sense that the user does not pay for it and Scilab is an open source software, provided under the Cecil license. The software is distributed with source code, so that the user has an access to Scilab most internal aspects. Most of the time, the user downloads and installs, a binary version of Scilab since the Scilab consortium provides Windows, Linux and Mac OS executable versions. An online help is provided in many local languages. From a scientific point of view, Scilab comes with many features. At the very beginning of Scilab, features were focused on linear algebra. But, rapidly, the number of features extended to cover many areas of scientific computing. The following is a short list of its capabilities:-

- Linear algebra, sparse matrices,
- Polynomials and rational functions,
- Interpolation, approximation,
- Linear, quadratic and non-linear optimization,
- Ordinary Differential Equation solver and Differential Algebraic
- Equations solver,
- Classic and robust control, Linear Matrix Inequality optimization,
- Differentiable and non-differentiable optimization,
- Signal processing,
- Statistics

1.2 What is Scilab?

Scilab is a free and open source software for engineers & scientists, with a long history (first release in 1994) and a growing community (100 000 downloads every months worldwide).

Scilab is a free and open-source, cross-platform numerical computation package and a high level, numerically oriented programming language. It can be used for signal processing, statistical analysis, image enhancement, dynamics simulations, numerical optimization, and modelling, simulation of explicit and implicit systems and (if the corresponding toolbox is installed) symbolic manipulations.\

Scilab is a high-level, numerically oriented programming language. The language provides an interpreted programming environment, with matrices as the main data type. By using matrix based computation, dynamic typing, and automatic memory management, many numerical problems may be expressed in a reduced number of code lines, as compared to similar solutions using traditional languages, such as Fortran, C, or C++. This allows users to rapidly construct models for a range of mathematical problems. While the language provides simple matrix operations such as multiplication, the Scilab package also provides a library of high level operations such as correlation and complex multidimensional arithmetic. The software can be used for signal processing, statistical analysis, image enhancement, fluid dynamics simulations, and numerical optimization.

Scilab also includes a free package called Xcos (a fork of Scicos based on Multilanguage) for modeling and simulation of explicit and implicit dynamical systems, including both continuous and discrete sub-systems. Xcos is the open source equivalent to Simulink from the Math Works.

As the syntax of Scilab is similar to MATLAB, Scilab includes a source code translator for assisting the conversion of code from MATLAB to Scilab. Scilab is available free of cost under an open source license. Due to the open source nature of the software, some user contributions have been integrated into the main program.

1.3 Features of Scilab:-

Scilab has main features that enable users interact more and easily with Scilab. They include optimization, statistics, maths and simulation, signal processing, application development, 2D and 3-D visualization and the control system design and analysis. Scilab includes hundreds of mathematical functions. It has a high level programming language allowing access to advanced data structures, 2-D and 3-D graphical functions. A large number of functionalities is included in Scilab: control, simulation, optimization, signal processing..., and Xcos, the hybrid dynamic systems modeler and simulator is provided with the platform

Scilab has been designed to be an open system where the user can define new data types and operations on these data types by using overloading. A large number of contributions can be downloaded from Scilab website. Scilab has been built using a number of external libraries. Scilab has been designed to be an open system where the user can define new data types and operations on these data types by using overloading. A large number of contributions can be downloaded from Scilab website.

Scilab has been built using a number of external libraries. It works on most UNIX systems (including GNU/Linux), Windows (9X/2000/XP/Vista) and comes with source code, online help and English user manuals.

- 2-D and 3-D Graphics, Animation
- Simulation: ODE solver and DAE solver
- Signal Processing
- Image Processing
- Parallel Scilab
- Instruments Modellin and Control
- Data analysis and Statistics, Data Handling
- Serial Communication
- Education
- Real-time simulation
- Number Theory and numerical math's

1.4 What is the purpose of Scilab?

Scilab includes hundreds of mathematical functions. It has a high level programming language allowing access to advanced data structures, 2-D and 3D graphical functions. For usual engineering and science applications including mathematical operations and data analysis.

1.5 Image Processing:

1.5.1 What is Image?

Image An image is an array, or a matrix, of square pixels (picture elements) arranged in columns and rows .An Image is a 2D function f(x, y) , where x and y are spatial coordinates and amplitude off at any pair of coordinates (x, y) is called the intensity or grey level of the image.

- An image is nothing, but more than two dimensional signals.
- It is defined by the mathematical function, f(x,y)
  Where,
  X co-ordinates is a Horizontal.
  Y co-ordinates is a Vertical.

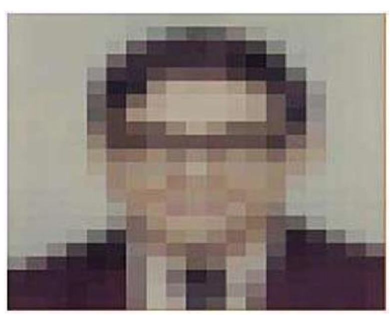


Figure 1.1 an image: an array or matrix of pixels arranged in columns and rows

1.5.2 Types of Images –

1) **Binary Image :** Are the simplest types of images and they take discreet values either 0 or 1 hence called binary images. Black is denoted by 1 and white by 0. These images have application in computer vision and used when only outline of the image required.

2) **Gray scale images :** They are also known as monochrome images as they do not represent any color only the level of brightness for one color. This type of image consists of only 8 bytes that is $256(0 - 255)$ levels of brightness 0 is for black and 255 is white in between are various levels of brightness.

3) **Colored images :** Usually consist of 3 bands red green and blue each having 8 bytes of intensity. The various intensity levels in each band is able to convey the entire colored image it is a 24 bit colored image.

1.5.3 What are the Characteristic Properties of Images?

A 2D digital image consists of a finite number of points aligned as rows and columns. A point of a digital image is called “pixel”. “Pixel” means “picture element”. At each pixel there can be a scalar gray value or a vector of color components or gray values. In Scilab there are the following types of images

Gray value images, also called “gray scale” images or “intensity images”: A gray value image is a matrix of gray values. In Scilab a gray value image is a 2D array. A gray value is usually an integer scalar between 0 and 255 or a real number between zero and one.

Pseudo color images, also called “indexed images”: Each pixel of a pseudo color image corresponds to an item in a list of colors. A list of colors is called “color map”. A color map is a matrix. The rows of this matrix correspond to color vectors and the columns correspond to color channels.

Color images: At each pixel of a color image there is a vector of color components, e. g. red - green - blue. Mathematically a color image can be described as a triple of matrices. Each matrix corresponds to a color channel. In Scilab, color images are represented as 3D arrays. The first dimension corresponds to the rows, the second dimension corresponds to the columns and the third dimension corresponds to the color channel. A color component is usually an integer scalar between 0 and 255 or a real number between zero and one. A color image can be transformed to a gray value image by calculating the scalar product or each color vector and a constant vector.

Logical images, also called “binary images”: A logical image is a matrix of Boolean values. When a logical image is visualized, false is displayed in black and true is displayed in white. Logical images can be generated comparing each pixel of a gray value image to a threshold. Pixels with gray values at least as high as the threshold are mapped to true whereas pixels with gray values lower than the threshold get mapped to false.

Object images, also called “label images”: An object image consists of objects and background. At each object image there is a number greater than zero. Pixels with the same number belong to the same object. Pixels with different numbers belong to different objects. Background pixels have the value zero. An object image can be created searching connected areas in logical images. Object images can be visualized as pseudo color images.

1.6 Histogram of the Image:

1.6.1 What is Histogram?

An image histogram is a chart that shows the distribution of intensities in an indexed or grayscale image. You can use the information in a histogram to choose an appropriate enhancement operation. For example, if an image histogram shows that the range of intensity values is small, you can use an intensity adjustment function to spread the values across a wider range.

A histogram is a graph. A graph that shows frequency of anything. Usually histogram have bars that represent frequency of occurring of data in the whole data set.

A Histogram has two axis the x axis and the y axis.

The x axis contains event whose frequency you have to count.

The y axis contains frequency.

An image histogram is a type of histogram that acts as a graphical representation of the tonal distribution in a digital image. It plots the number of pixels for each tonal value. By looking at the histogram for a specific image a viewer will be able to judge the entire tonal distribution at a glance.

Image histograms are present on many modern digital cameras. Photographers can use them as an aid to show the distribution of tones captured, and whether image detail has been lost to blown-out highlights or blacked-out shadows. This is less useful when using a raw image format, as the dynamic range of the displayed image may only be an approximation to that in the raw file.

The horizontal axis of the graph represents the tonal variations, while the vertical axis represents the number of pixels in that particular tone.The left side of the horizontal axis represents the black and dark areas, the middle represents medium grey and the right hand side represents light and pure white areas. The vertical axis represents the size of the area that is captured in each one of these zones. Thus, the histogram for a very dark image will have most of its data points on the left side and centers of the graph. Conversely, the histogram for a very bright image with few dark areas and/or shadows will have most of its data points on the right side and center of the graph.

Image histograms look like those terrible mathematical graphs we were so happy to forget after graduating! Yes, they are graphs, but they are very easy to interpret. And they are extremely helpful in photography once you learn some very simple basics.

Image histogram is a graph plotting the frequency of occurrence of different color intensities in the image. Simply put, it shows how many pixels of every possible color there are in the image. Every bar on the image histogram represents one intensity level. 0 (black) is usually shown on the left, and 255 (white) on the right. Look at how high the bar is, and you see how many image pixels have the corresponding color. Let me give some examples.

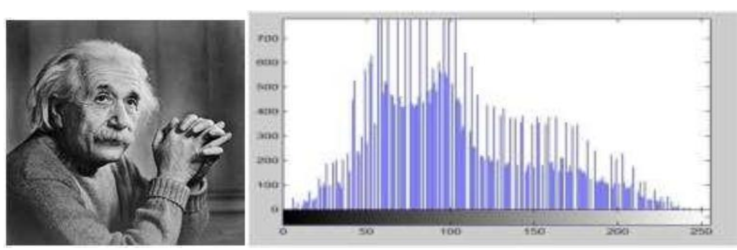


Fig 1.2 Image and there Histogram

1.6.2 Application of histogram

- In digital image processing, the histogram is used for graphical representation of a digital image.

- In digital image processing, histograms are used for simple calculations in Software.
- It is used to analyze an image. Properties of an image can be predicted by The detailed study of the histogram.
- The brightness of the image can be adjusted by having the details of its Histogram
- The contrast of the image can be adjusted according to the need by having Details of the x-axis of a histogram.
- It is used for image equalization. Gray level intensities are expanded along The x-axis to produce a high contrast image.
- Histogram are used in the holding as it improves the appearance the Image**.**

Chapter 2: Literature Survey

2.1 Commands use in Image Processing

1. **imread -** Reads image file in any format.

   Syntax – Image = imread (filename)
   [Index, Map] = imread (filename)

2. **imwrite –** Write image to file in any format.

   Syntax - imwrite(Image, filename, [opt_args])
   imwrite(Index, Map, filename, [opt_args])

3. **imshow -** Displays images in the Scilab graphic window.

   Syntax - imshow(img)

4. **imhist -** imhist return the histogram of an image. If more than 2 arguments are given, the histogram will be shown in a figure.

   Syntax - [count, cells] = imhist(im)
   [Count, cells] = imhist(im, bins)
   [Count, cells] = imhist(im, bin [, width [, color]])

   e.g. –

   ```
   m = imread('image.jfif')
   im = rgb2gray(m);
   [Count, cells]=imhist (im);
   ```

```
[count, cells]=imhist(im, 10);
scf(1); imhist(im, 10, 0.5); scf(2);
imhist(im, 10, 'green'); scf(3);
imhist(im, 10, 0.8, 'green');
```

5. **imadjust -** This function use to adjust the intensity of an image using histogram range method. The new image would be map into the new range with given min and max values.

   Syntax - imout = imadjust(imin, src, dest)

   e.g. –

```
img = imread('image.jfif');
gray = rgb2gray(img);
adj_img = imadjust(gray, [0.3,0.7],[]);
subplot(121);
imshow(gray);
title('input');
subplot(122);
imshow(adj_img);
title('adjusted');
```

6. **imnoise -** Add noise (gaussian, etc.) to an image

   Syntax - imnoise(im, type [, parameters])

   e.g. -

```
im = imread('image.jfif');
imn = imnoise(im, 'gaussian');
imshow(imn);
imn = imnoise(im, 'salt & pepper');
```

```
imshow(imn);
imn = imnoise(im(:,:,1), 'salt & pepper', 0.2);
imshow(imn);
lowtri = tril(ones(im(:,:,1)));
imn = imnoise( im(:,:,1), 'localvar', lowtri/5);
imshow (imn);
imn = imnoise( im(:,:,1), 'localvar', [0:0.1:1], [0:0.1:1].^3);
imshow (imn);
imn = imnoise(im, 'speckle' ); imshow (imn);
```

7. **imcrop -** Crop image at regin rect to subim.

   Syntax - subim = imcrop(im, rect)

   e.g. –
   ```
   im = imread('image.jfif');
   subim = imcrop(im, [50, 60, 150, 100]);
   imshow(subim);
   ```

8. **imcropm -** This function allows user to select the region of interest interactively using mouse.

   Syntax - y = imcropm(S)

   e.g. -
   ```
   S = imread('image.jfif');
   J = imcropm(S);
   figure(); imshow(J);
   ```

9. **Bwborder -** Extracts contours from binary images, by detecting which pixel valued 1 has at least one neighbor valued 0.

Syntax - outm = bwborder(inm, conn)

e.g. -

```
inm = imread('image.jfif');
outm = bwborder(im2bw(inm,0.5), 4)
imshow(outm)
```

10. **im2bw -** im2bw convert intensity or RGB images to binary images. The output is a boolean matrix, which has value of %T for all pixels in the input image with luminance grater than thresh and %F for all the other pixels.

Syntax – im2 = im2bw(im, thresh)

e.g. –

```
S = imread('image.jfif');
S2 = rgb2gray(S);
Sbin = im2bw(S2,0.5);
imshow(Sbin);
```

2.2 Creating A Histogram

To create an image histogram, use the imhist() function. This function creates a histogram plot by making n equally spaced bins, each representing a range of data values. It then calculates the number of pixels within each range.

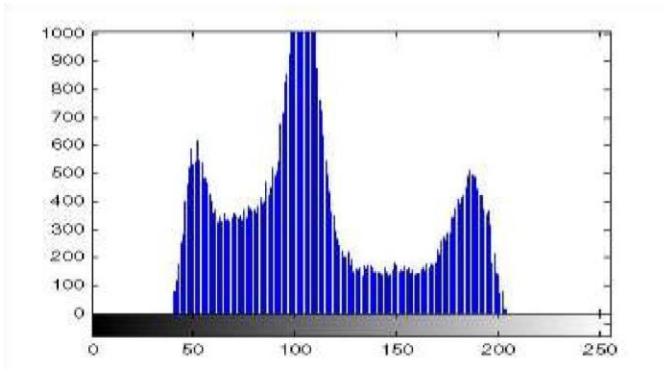
The following example displays an image of grains of rice and a histogram based on 64 bins. The histogram shows a peak at around 100, corresponding to the dark gray background in the image. For information about how to modify an image by changing the distribution of its histogram, see Adjusting Intensity value Specified Range.

1. Read image and display it.

- I = imread('rice.png');
- imshow(I)

2. Display histogram of image.

- figure,imhist(I)



2.3 Installation of Scilab

Whatever your platform is (i.e. Windows, Linux or Mac), Scilab binaries can be downloaded directly from the Scilab homepage.

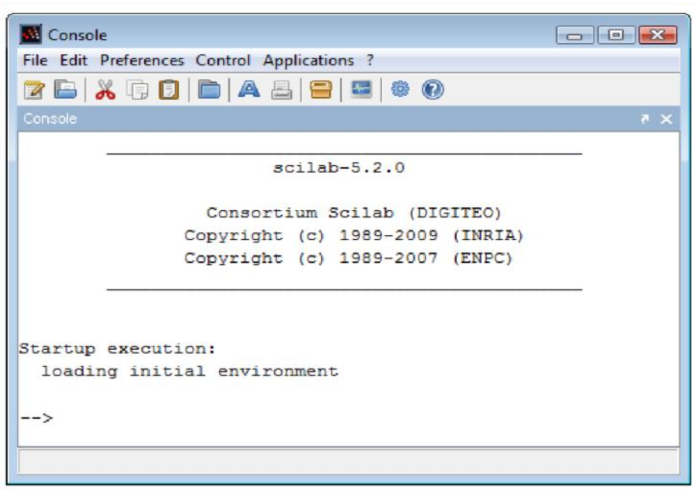


Figure 2.2: Scilab console under Windows

2.3.1 Installing Scilab under Windows

Scilab is distributed as a Windows binary and an installer is provided so that the installation is really easy. On Windows, if your machine is based on an Intel processor, the Intel Math Kernel Library (MKL) enables Scilab to perform faster numerical computations.

A lot of the articles on x-engineer.org are using Scilab as programming/simulation environment. In this article we are going to follow step by step the installation process of Scilab under Windows.

For this particular example we are using Scilab 5.4.0 (Beta 1 release) and Windows Vista. The same installation procedure should be valid for newer versions of Scilab and Windows.

Before you start the Scilab installation process make sure that you have administrator rights on Windows and an internet connection. Scilab is available for download at www.scilab.org

Windows Vista, 7, 8, 10

Scilab 6.0.2 - Windows 64 bits, (scilab-6.0.2_x64.exe - 181M) (exe)

Scilab 6.0.2 - Windows 32 bits, (scilab-6.0.2.exe - 172M) (exe)

Scilab 6.0.2 is a Latest version of Scilab.

Scilab 6.0.2 is a Released on Thu, 14 Feb 2019 under the terms of the GNU General Public License (GPL) v2.0.

System Requirements: Windows:

OS

- Windows Vista (32 and 64 bits)
- indows 7 (32 and 64 bits)
- Windows 8 (32 and 64 bits)
- Windows 10 (32 and 64 bits)

Hardware

- Pentium IV class (or equivalent) with SSE2 instructions is required
- 2 GB RAM (1 GB minimum)
- 600 MB hard disk space

Optional

- An Internet connection for ATOMS modules install (using a proxy requires manual configuration of ATOMS)

Chapter 3 Scope of the Project

3.1 What you get?

3.1.1 Expertise :

your concerns directly taken care of by the Scilab expert team. The best answers by the experts who are building the software.

3.1.2 Reliability :

the best of both worlds, open-source with enterprise-class professionals to rely on.

3.1.3 Reactivity :

your R&D problems cannot wait for the community. We guaranty the response time.

3.2 What does it include?

- A defined number of level-1-support hours
- Your answer within two business days (opening hours from 9 am to 5 pm CEST)
- One year long (no automatic renewal)
- For your whole team: no limitation of users
- Scale up your use of Scilab, scale your support

Chapter 4: Methodology

4.1 Advantages of Scilab Over MATLAB:

Not only is the obvious advantage of being open-source a big plus, but it also has the power to be improved by individual users as necessity arises; its structure is completely transparent to the user. Scilab is one of the two major open-source alternatives to MATLAB, the other one being GNU Octave. Scilab puts less emphasis on syntactic compatibility with MATLAB than Octave does, but it is similar enough that some authors suggest that it is easy to transfer skills between the two systems.

Scilab

The most popular points invoked when promoting Scilab are:-

1. **The cost**

   . Scilab can be downloaded for free and can be used on any computer in a company, regardless of its size. On the contrary, Matlab is regularly deemed by industrialists as being too expensive. To manage Matlab licenses within a firm adds a non-negligible expense

2. **The openness**

   Scilab is Open Source software. Every user has complete access to the source code, a crucial point when it comes to depending on an external supplier.

3. **Dynamism**

   the dynamism of the Scilab community which is an innovation criteria (implementation of new algorithms). In the case of a commercial solution, users depend on the choices of the publisher

4. **The quality**

   the algorithms available with Scilab are well known for their quality, with a level at least comparable with its competitor, which is great for freeware.

5. **Good guidance**

   Scilab is supported by an industrial consortium of which some of the companies are major ones (CEA, CNES, Dassault, EADS, EDF, PSA Peugeot Citroën, Renault, Thales). This aspect ensures the good guidance of future releases to users.

4.2 Scilab Disadvantages

- Numeric computing introduces rounding errors, contrary to symbolic computing
- The learning effort required by numeric computing is higher than for symbolic computing
- Scilab lacks a unified tutorial and/or user’s manual. You “try and cry” and waste time searching for information on its use.
- In some cases Scilab executes much slower than MATLAB and GNU Octave(improvements are said to be under way)
- Scilab’s tools for creating GUIs are poor compared with MATLAB
- The Help Browers is very formal and of little use to newbies
- Scilab has bugs and tends to Crash/lockup

4.3 Action plan

5th Semester Action Plan			
Week	Task	Start Date	Finish Date
1	Group discussion and selection of topic	26/07/19	01/08/19
2	• Select Title for the project • Research and read more about topic • Gathering information, group discussion and distributions of works to be done	02/08/19	08/08/19
3	Gathering information about existing system (Literature review and Industrial Survey)	09/08/19	14/08/19
4	Find our drawbacks of existing system	16/08/19	22/08/19
5	Study the Scilab tools(Software).	23/08/19	24/08/19
6	Study the Features of Scilab. Study the Concept of Scilab.	29/08/19	29/08/19
7	Study the the applications of Scilabsuch as Image processing etc.	30/08/19	05/09/19
8	Preparing UML Diagrams – Use case diagram, Activity diagram, DFD diagram, ER diagram.	06/09/19	11/09/19
9	Preparing the project report	13/09/19	19/09/19

6th Semester Action Plan			
Week	Task	Start Date	Finish Date
1	• Download setup of cilab. • Installation of Scilab Properly.	12/12/19	19/12/19
2	• Observe the Scilab tools. • Familiar to the Scilab tools.	20/12/19	26/12/19
3	Develop the code for Basics.	27/12/19	02/01/20
4	Develop code for basic .	03/01/20	09/01/20
5	Develop the code for basic Mathematical Operations for practice.	10/01/20	16/01/20
6	Develop the code for Image Processing on Scilab.	17/01/20	23/01/20
7	Develop the code for Histogram of the image.	24/01/20	30/01/20
8	Observe the Output of the Histogram.	31/01/20	06/02/20
9	Finalizing the Report on Scilab.	07/02/20	13/02/20
10	Submitted the Report on Teacher.	14/02/20	20/02/20

Chapter 5: Details of design, working and processes

5.1 UML Diagrams

5.1.1 Use Case Diagram

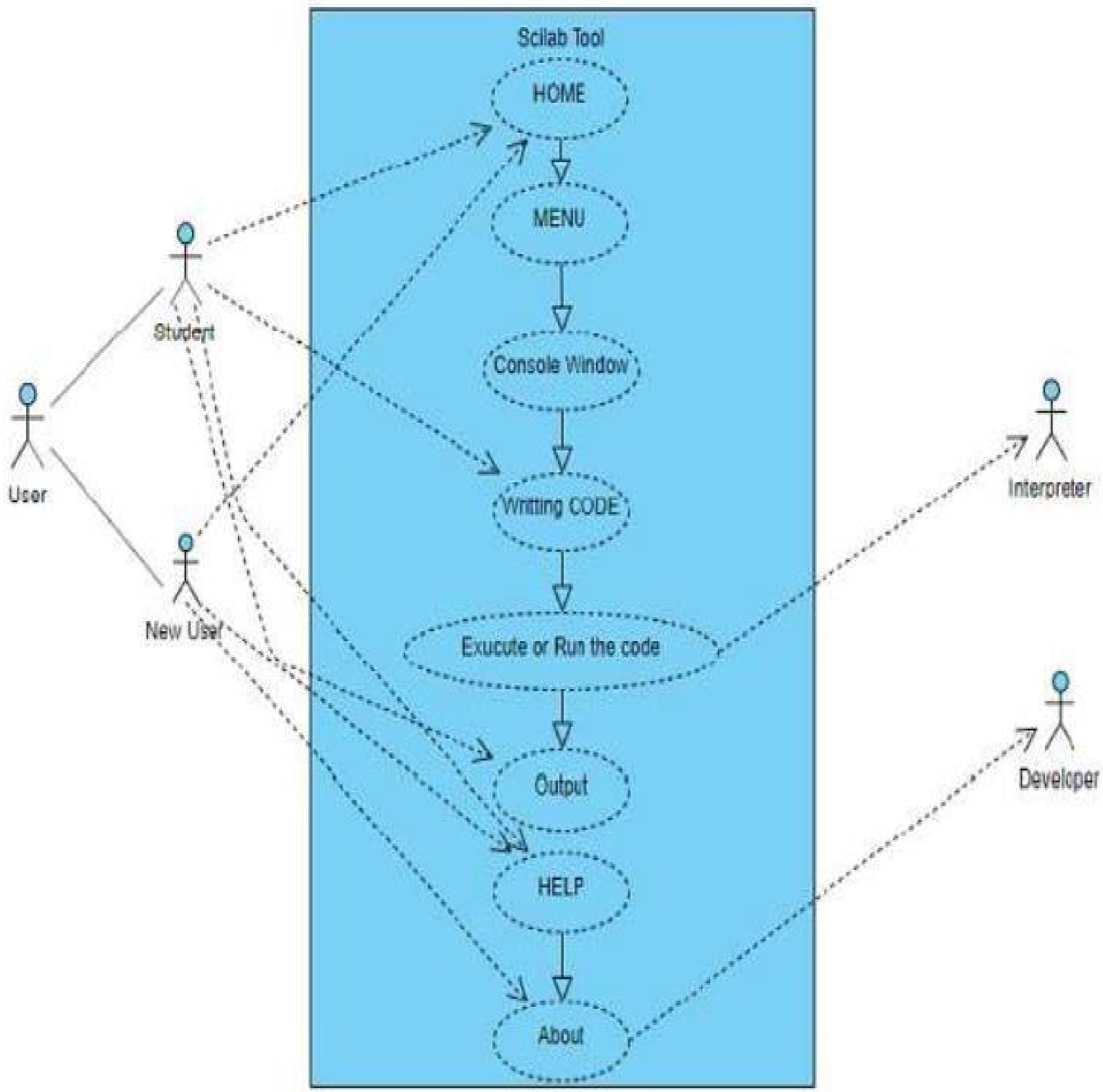


Diagram 5.1

5.1.2 Activity Diagram

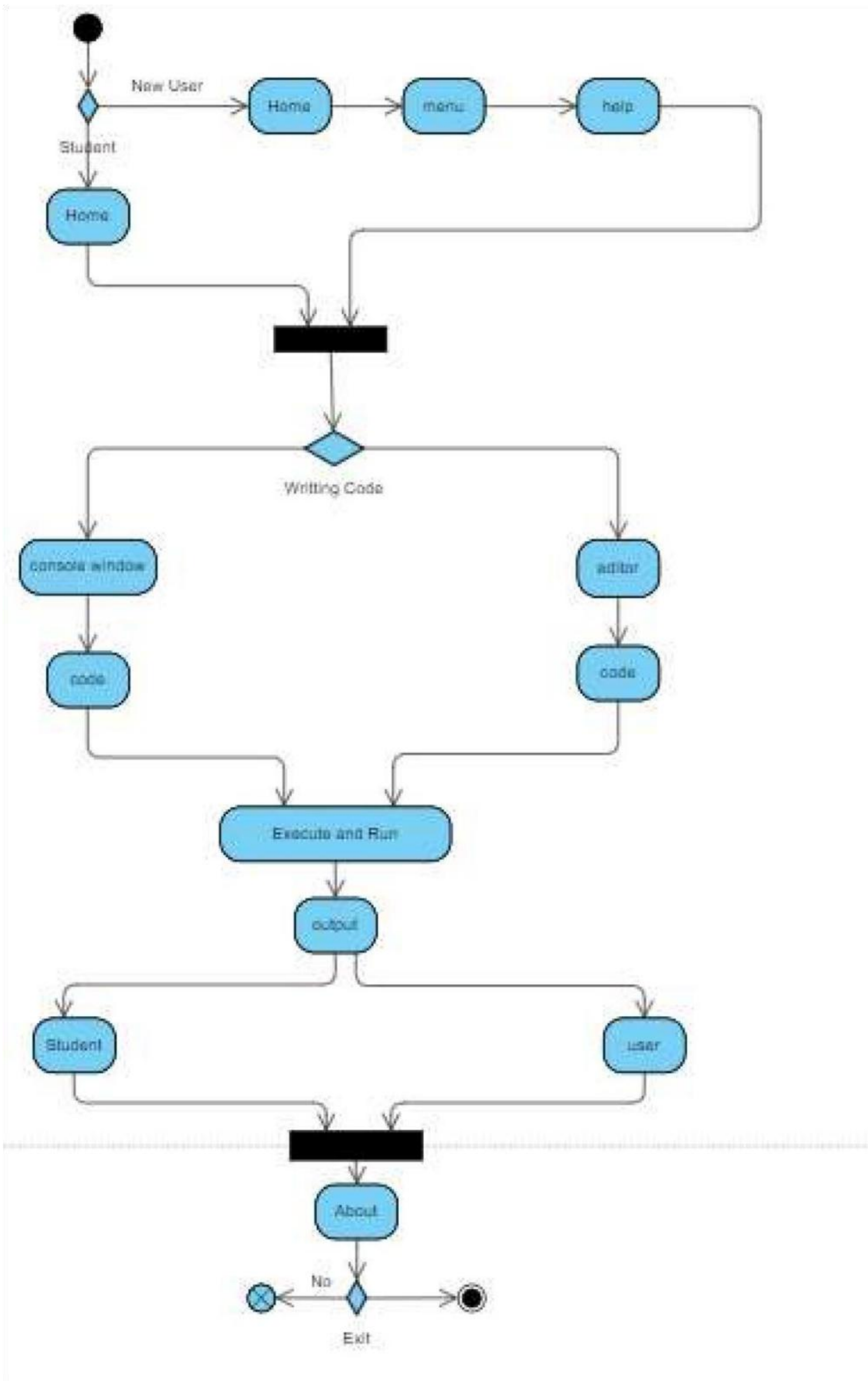


Diagram 5.2

6.1.3 E-R Diagram

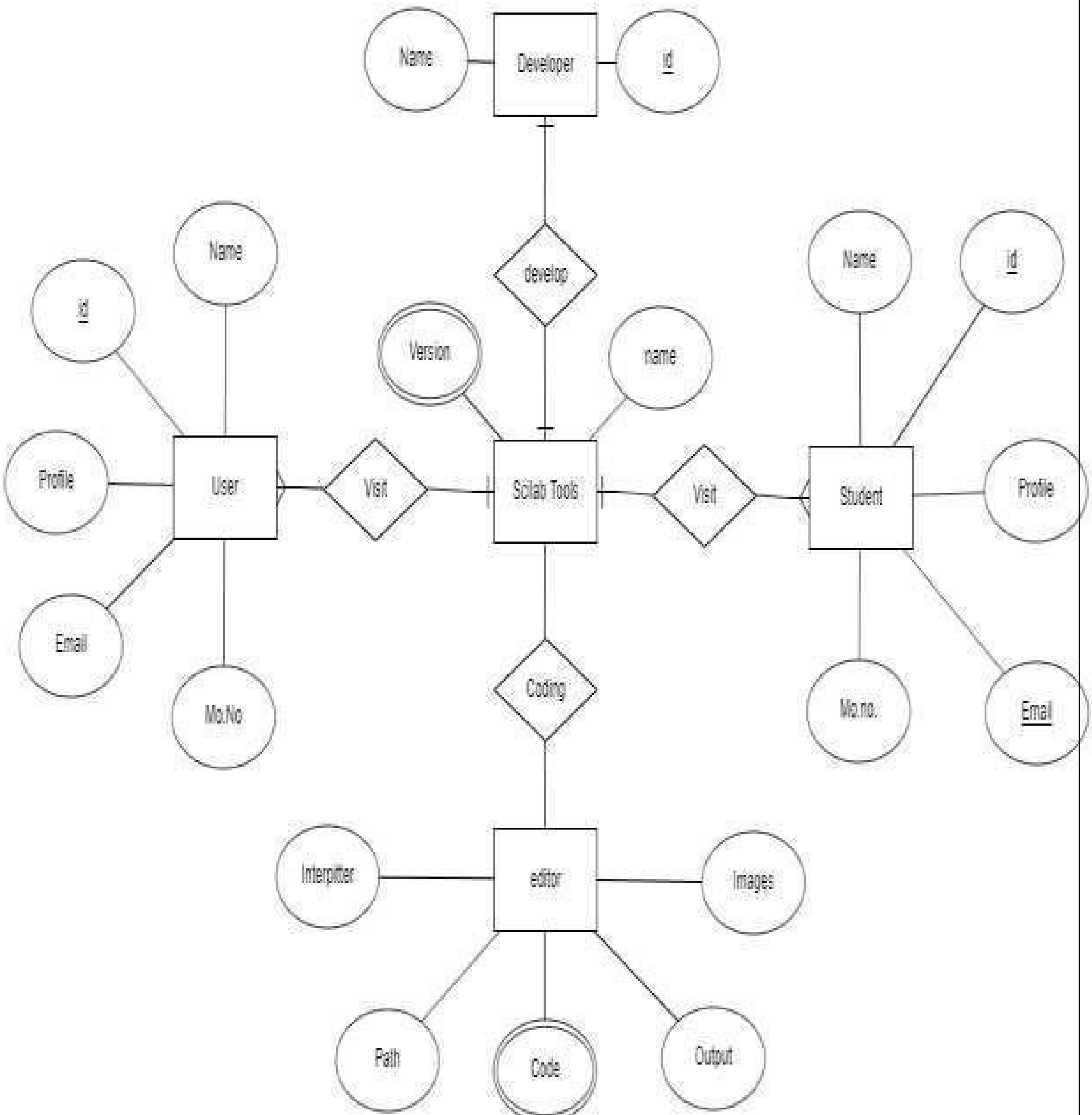
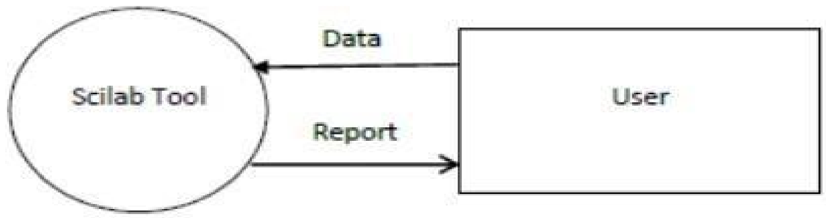
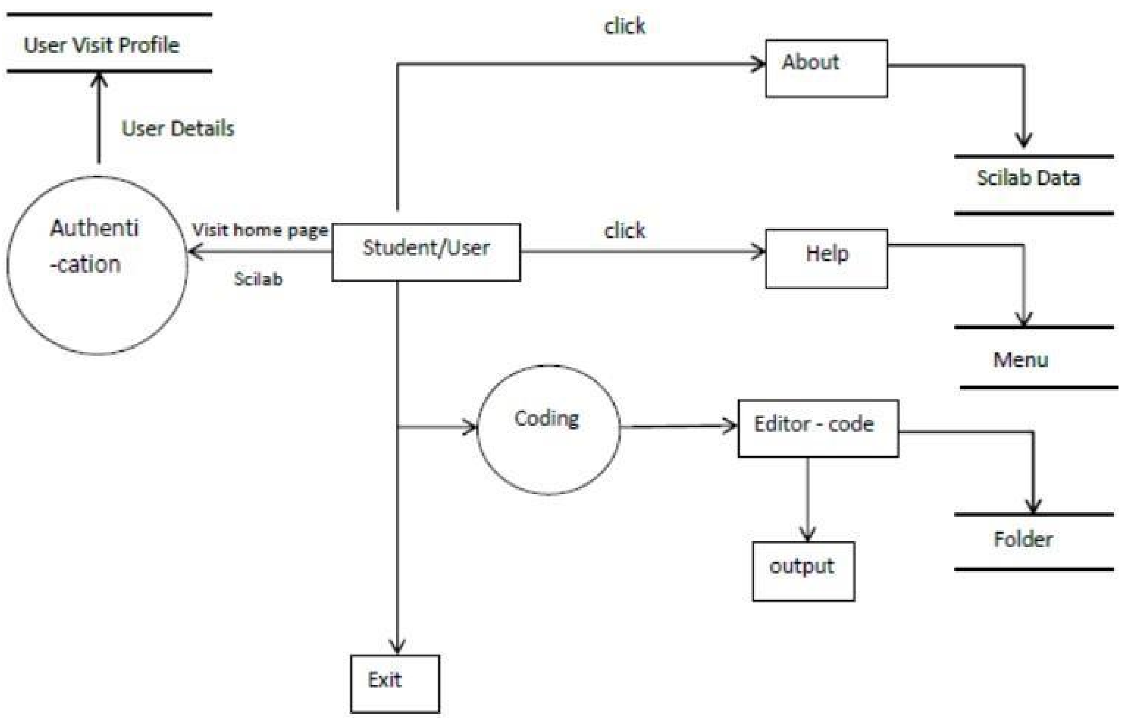


Diagram 5.3

6.1.4 Data flow Diagram:



Level 0 DFD



Level 1 DFD

Diagram 5.4

5.2 SYSTEM TESTING:

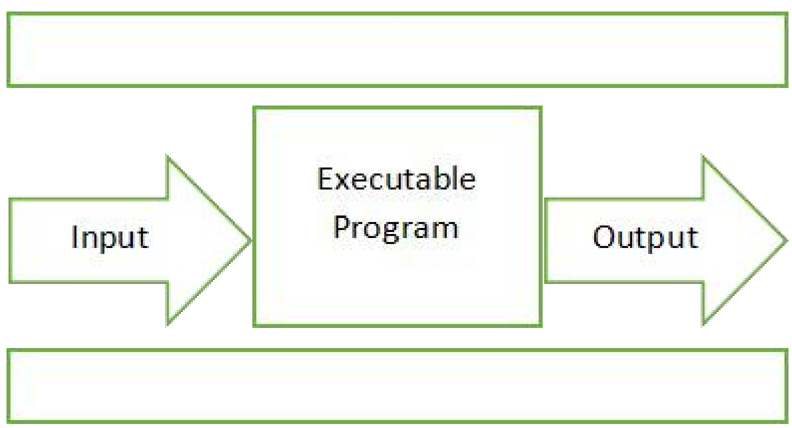
- **WHITE-BOX TESTING**

In white-box testing the software tester has access to the program's code and can examine it for clues to help with his testing he can see inside the box.

White-box testing is also structural testing or glass box tesing. White-box testing is performed to test the program internal testing.

The goal of white-box testing is to ensure that the test case(developed by software testers by using white-box testing) exercise each path through a program. That is, test cases ensure that all internal structures in the program are developed according to design specifications.

Following shows white-box testing.



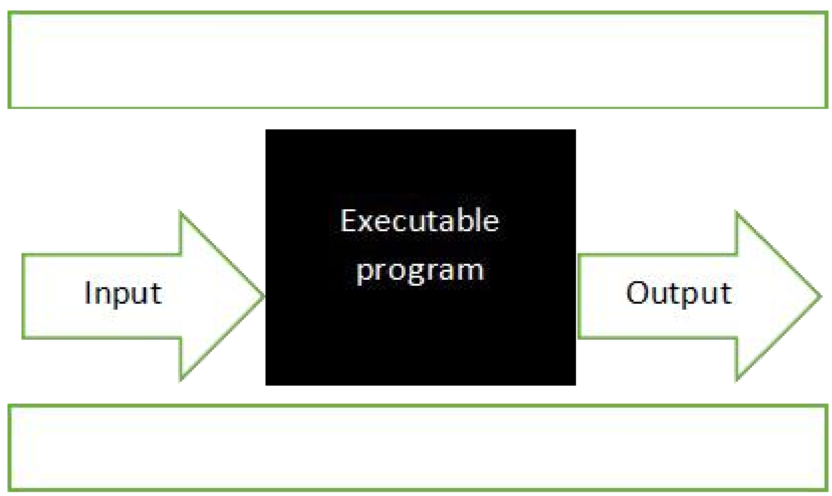
- **BLACK-BOX TESTING**

Black-box (or functional) testing checks the functional requirements and examines the input and output data of these requirements

Once, the specified function for which the software has been designed is known, tests are performed to ensure that each function is working properly. This is referred to as black-box testing.

Black-box testing also known as Behavioural Testing, is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. It is a software testing method in which the internal structure /design/implementation of item being tested is not known to the tester.

Following figure black-box testing.



- **Unit Testing:**

Unit Testing is a level of the software testing process where individual units/components of software/system are tested. The purpose is to validate that each unit of the software performs as designed.

Testing that occurs at the lowest level is called unit testing is called module testing. Unit testing performed to test the individual units of software. Unit is the smallest part of software system which is testable.

Unit testing may be include code file, classes, methods which can be tested individually for correctness.

The primary goal of unit tesing is to take the smallest piece of testable software in the application, isolate it from the remainder of the code, and determine whether it behaves exactly as you expect.

- **Smoke Testing:**

Smoke testing involves testing basic functionality of software application developed to ensure that application is living and one can work on it. Generally, smoke tests are performed without any user input.

Smoke testing term came from hardware testing, when you get new hardware and power it on if smoke comes out then you do not proceed with testing.

Smoke testing is done to check the normal health of the build and make sure if it is possible to continue testing. Smoke testing is an integration testing approach that is used when software product is developed.

- **Sanity Testing:**

Sanity Testing is performed to test the major functionality or behaviour of the software application.

Sanity testing is done after thought regression testing is over, it is done to make sure that any defect fixes or changes after regression testing does not break the core functionality of the product. It is done towards to end of the product release phase.

Sanity testing follows narrow and keep approach with detailed testing of some limited features.

Sanity testing is like doing some specialized testing which is used to find problems in particular functionality.

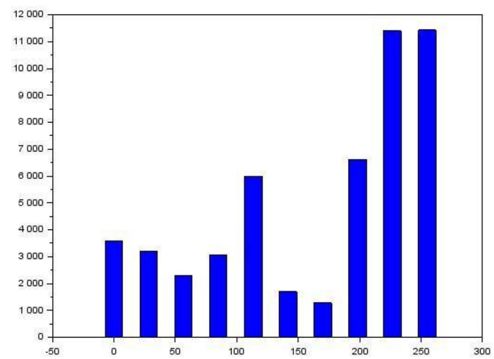
Sanity Testing is done with an intent to verify that end user requirements are met on not.

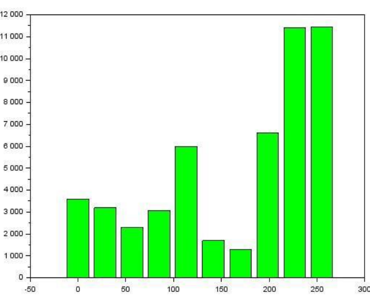
Sanity Testing are mostly non scripted.

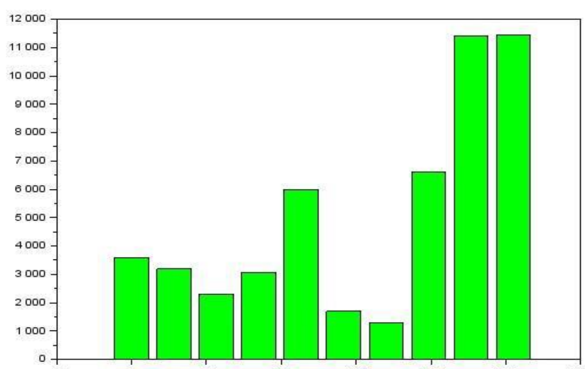
Chapter 6: Results and Application

6.1 Outputs of Commands:

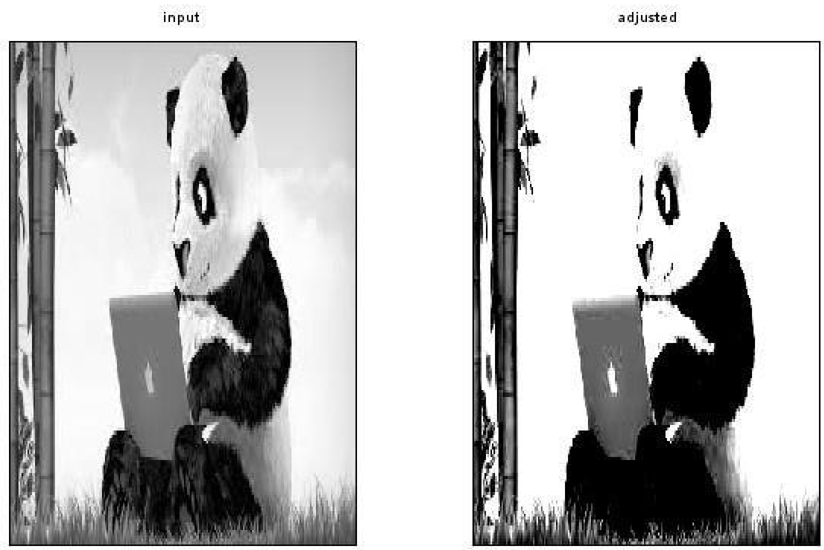
1. **imhist –**







2. **imadjust –**

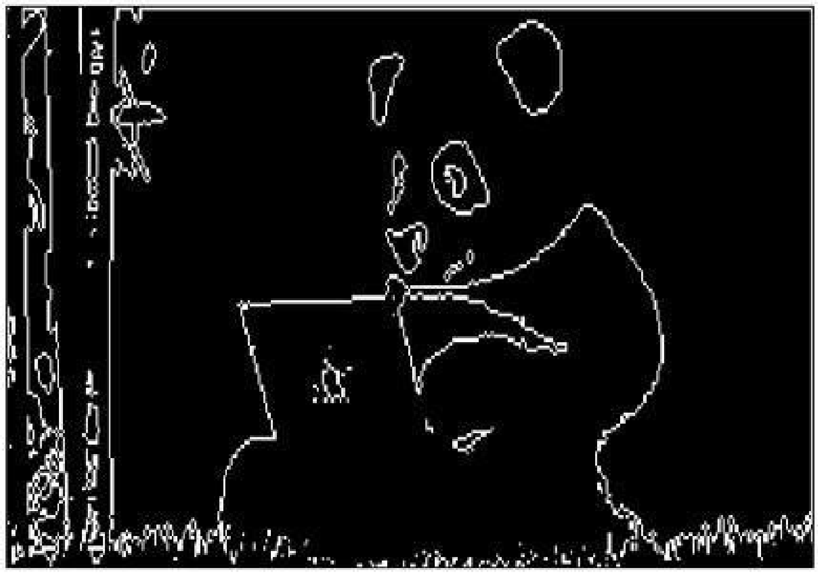


3. **imnoise –**

4. **imcrop –**

5. **imcropm –**

6. **bwborder**



7. **im2bw**

Chapter 7: Conclusions and future scope

7.1 Conclusions:

Hence from this project We have learn the Scilab tool(Software) properly. We have also learn the one of the application of scilab , as image processing. We study the commands use in image processing such as –imhist, imread, imshow, imwrite, imfinfo . We studies the what is the image, purpose of scilab, what is the histogram. We study the concept of scilab and the features of scilab , similar as learn the Advantage and Disadvantage of scilab. In last We learn the how to create and draw the histogram of the image. We also perform coding on scilab tools and observe the Outputs.

7.2 ` Future scope:

The future of image processing will involve scanning the heavens for other intelligent life out in space. Also new intelligent, digital species created entirely by research scientists in various nations of the world will include advances in image processing applications. Due to advances in image processing and related technologies there will be millions and millions of robots in the world in a few decades time, transforming the way the world is managed. Advances in image processing and artificial intelligence will involve spoken commands, anticipating the information requirements of governments, translating languages, recognizing and tracking people and things, diagnosing medical conditions, performing surgery, reprogramming defects in human DNA, and automatic driving all forms of transport. With increasing power and sophistication of modern computing, the concept of computation can go beyond the present limits and in future, image processing technology will advance and the visual system of man can be replicated. The future trend in remote sensing will be towards improved sensors that record the same scene in many spectral channels. Graphics data is becoming increasingly important in image processing applications. The future image processing applications of satellite based imaging ranges from planetary exploration to surveillance applications.

Chapter 8: References and Bibliography

[1] INTRODUCTION TO SCILAB.(Consortium scilab)

[2] Scilab Manual For Digital Image Processing (Dr.B.Chandra Mohan college)

[3] Study of color Histogram Based Image Retrival (Rishav Chakravarti).

[4] Image Processing with Scilab and Image Processing Design Toolbox Copyright c by Dr. Eng. (J) Harald Galda, 2011

[5] https://www.scilab.org

[6] https://www.wikipedia.org/

[7] https://erdplus.com/standalone

[8] Visual paradigm online Diagram http://online.visual paradigm .com.driver/ diagramlist:proj=0new.

[9] www.latex-project.org

[10] YouTube video –

https://www.youtube.com/watch?v=yT3zEcNMhWE
https://www.youtube.com/watch?v=TTFFZFCJFM4