IMAGE MODIFIER

A PROJECT REPORT

for

Mini Project (KCA353)

Session (2023-24)

Submitted By

AKANKSHA MANGAL
University Roll No- 2200290140015

Submitted in partial fulfillment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATIONS

Under the Supervision of
Ms. Divya Singhal
(ASSISTANT PROFESSOR)

Submitted to
DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206
(March 2024)

CERTIFICATE

Certified that **Akanksha Mangal 220029014007873** has carried out the project work having "**Image Modifier**" (**Mini Project-KCA353**) for **Master of Computer Application** from Dr.A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Akanksha Mangal
(2200290140015)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Ms. Divya Singhal
Assistant Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Dr. Arun Tripathi
Head
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

IMAGE MODIFIER

AKANKSHA MANGAL

ABSTRACT

Advancement in today's digital age, image editing has become an integral part of various industries, including photography, graphic design, and social media. This abstract presents an innovative Java-based image editing system designed to provide users with a versatile and user- friendly platform for enhancing and manipulating digital images. This system leverages the power of Java's object-oriented programming capabilities to offer a wide range of image editing functionalities, making it a valuable tool for both amateur and professional users.

The Java-based image editing system comprises a robust set of features, including image import/export, basic editing tools (such as cropping, resizing, and rotation), advanced filters and effects (such as blurring, sharpening, and color correction), and text overlay capabilities. The system's intuitive user interface ensures ease of navigation and accessibility for users of all levels of expertise. Additionally, it supports a variety of popular image file formats, ensuring compatibility with various image sources.

ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Ms. Divya Singhal** for her guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Arun Kumar Tripathi, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Akanksha Mangal
2200290140015

List of Chapters

CHAPTER 1

INTRODUCTION

Our project topic is “IMAGE PROCESSING TECHNIQUES”. It is a desktop-based application. This project aims at creating various effects for processing an image of any format such as .jpg, .gif etc. Our objective is to give a clear outlook about the various operations or effects that can give to an image to change its original look. We select this topic as our project by acquiring motivations from various existing software’s such as Windows Picture Management likewise…We use java net beans as a supporting software while commencing this project. The pixel grabber function in java helps to grab each image into its pixel level.

Image Processing is the art and science of manipulating digital images. It stands with one foot firmly in mathematics and the other in aesthetics and is a critical component of graphical computer systems. It is a genuinely useful standalone application of Java2D.The 2D API introduces a straight forward image processing model to help developers manipulate these image pixels. The image processing parts of Java are buried within the java.awt. image package.

1.1 PROJECT DESCRIPTION:

The Image Modifier is a software tool designed to provide users with the capability to modify and enhance digital images. Its primary purpose is to empower users to customize various aspects of an image, enabling them to achieve desired visual effects, correct imperfections, or align the image with specific requirements. This overview outlines key features, functionalities, and potential applications of the Image Modifier.

the Image Modifier is a versatile software solution that empowers users to manipulate digital images for various purposes. Whether for artistic endeavors, professional design projects, or personal image enhancement, the Image Modifier provides a user-friendly and feature-rich platform for effective image customization.

1.2 PROJECT SCOPE:

The project scope of an image modifier can vary based on specific requirements and goals, but generally, it involves creating a tool or application that can manipulate digital images in various ways.

The specific features and functionalities may vary depending on the intended use of the image modifier and the target audience.

1.3 HARDWARE/SOFTWARE USED IN PROJECT:

1.3.1 Hardware Requirements:

To implement a new system, the choice of a processor with maximum possible speed is made there should be sufficient memory to store data and software tools for efficient processing.

- **Processor:** Pentium IV
- **Clock Speed:** 800 MHz
- **Main Memory:** 256 MB RAM
- **Disk Storage:** 80 GB or above.
- **Monitor:** SVGA Color
- **Keyboard:** 108 Keys
- **Mouse:** Logitech
- **Floppy Disk Drive:** 3.5 floppy 1.44 MB
- **Compact Disk Drive:** 52 x

1.3.2 Software Requirements:

- **Operating System:** Windows 2000
- **Front End:** Java Net Beans

CHAPTER 2

FEASIBILITY STUDY

A feasibility study on an image modifier would typically involve assessing the technical, economic, legal, operational, and scheduling aspects of developing and implementing such a system.

This feasibility study delves into the technical, economic, legal, operational, and scheduling aspects of developing and implementing an image modifier. By understanding the implications and challenges associated with this innovative tool, stakeholders can make informed decisions regarding its development and integration into diverse workflows.

2.1 **TECHNICAL FEASIBILITY:**

- Evaluate the technical requirements and challenges associated with developing an image modifier.
- Assess the availability of technology, tools, and skills required for implementation.
- Consider potential technical risks and limitations.

2.2 **OPERATIONAL FEASIBILITY:**

- Assess how well the image modifier aligns with the existing workflow or operations.
- Consider the impact on users and stakeholders during and after implementation.
- Identify training needs and any potential resistance to adoption.

2.3 **BEHAVIORAL FEASIBILITY:**

- Develop a realistic timeline for the development and deployment of the image modifier.
- Consider dependencies, potential delays, and critical milestones.
- Assess whether the project can be completed within a reasonable timeframe.

2.4 **ECONOMIC FEASIBILITY:**

- Estimate the costs involved in developing and maintaining the image modifier.
- Evaluate the potential return on investment (ROI) or benefits derived from the system.

CHAPTER 3

DATABASE DESIGN

3.1 DATABASE TABLES:

Software design is the preliminary step and is also a building block of software. engineering. The efficiency of the software is promoted through the design phase. The design phase begins when the requirement specification document for the software to develop is available. Design is essentially the bridge between the requirement specifications and the final solution for satisfying the requirement. It is done in three stages such as external design, architectural design and detailed design. While the requirement specification activity is entirely in the problem domain, design is the first step to moving from the problem domain towards the solution domain. Design is essentially the bridge between the requirements. specification and the final solution for satisfying the requirements.

3.2 FLOW CHART:

The flowchart is a picture of the separate steps of a process in sequential order. It is a generic tool that can be adapted for a wide variety of purposes, and can be used to describe various processes, such as a manufacturing process, an administrative or service process, or a project plan.

flowchart for an image modifier involves a detailed breakdown of various functionalities and user interactions. Below is a simplified flowchart representing the basic workflow of an image modifier application:

plaintext

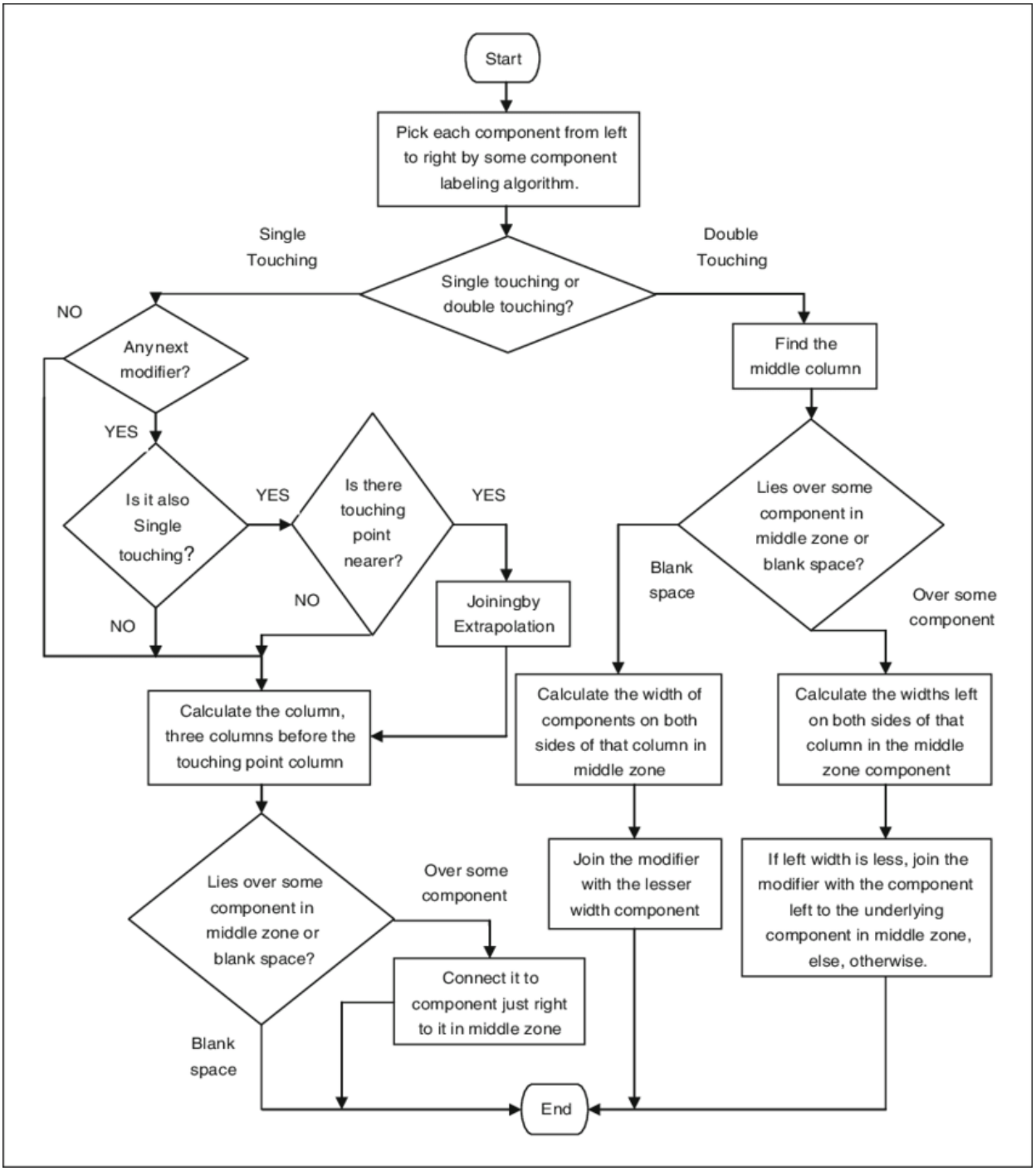
Start
Pick each component from left to right by some component labeling algorithm.
Single Touching
Double Touching
Single touching or double touching?
NO
Any next modifier?
YES
Is it also Single touching?
YES
Is there touching point nearer?
YES
NO
NO
Joiningby Extrapolation
Find the middle column
Lies over some component in middle zone or blank space?
Blank space
Over some component
Calculate the column, three columns before the touching point column
Calculate the width of components on both sides of that column in middle zone
Calculate the widths left on both sides of that column in the middle zone component
Lies over some component in middle zone or blank space?
Over some component
Join the modifier with the lesser width component
If left width is less, join the modifier with the component left to the underlying component in middle zone, else, otherwise.
Connect it to component just right to it in middle zone
Blank space
End

Fig: 3.2.1

3.3 USE CASE DIAGRAM:

A use case diagram is a visual representation of the interactions between users (actors) and a system in terms of use cases.

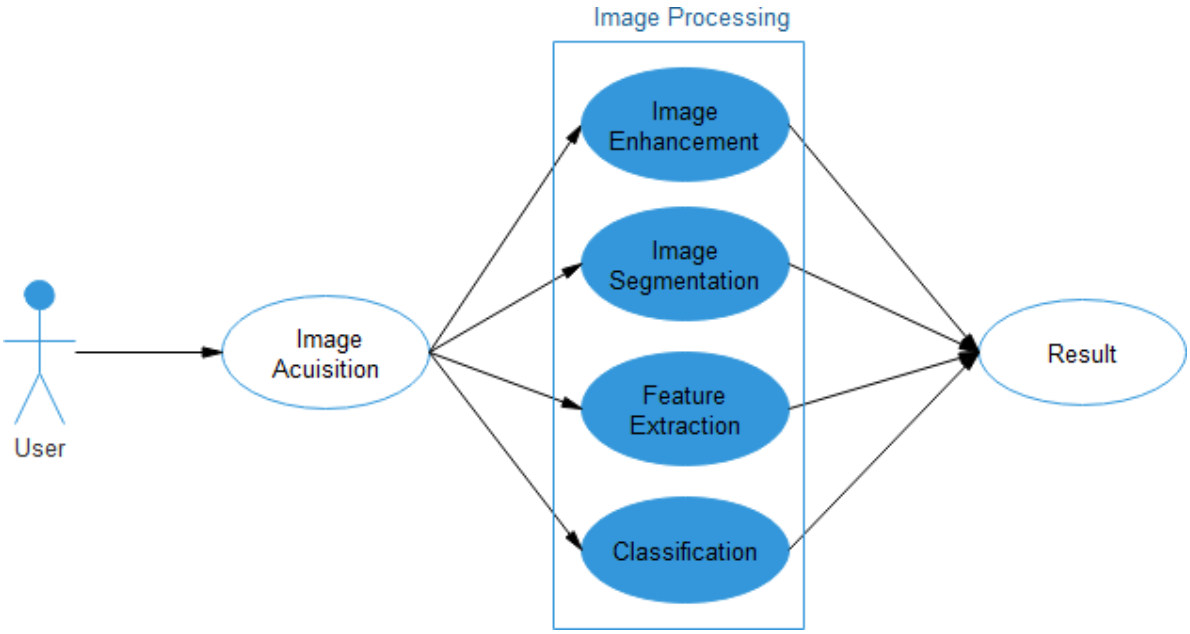

Fig: 3.3.1

3.4 SEQUENCE DIAGRAM:

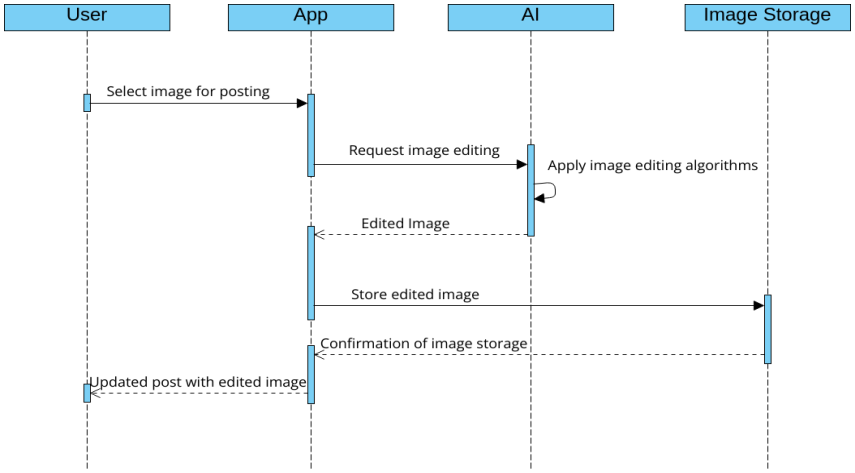

Fig: 3.4.1

3.5 COLLABORATIVE DIAGRAM:

A collaboration diagram, also known as a communication diagram, illustrates how objects collaborate to achieve a particular task or functionality. In the context of an image modifier, objects could include the user interface, image processing components, and other relevant entities. Here's a simplified collaboration diagram:

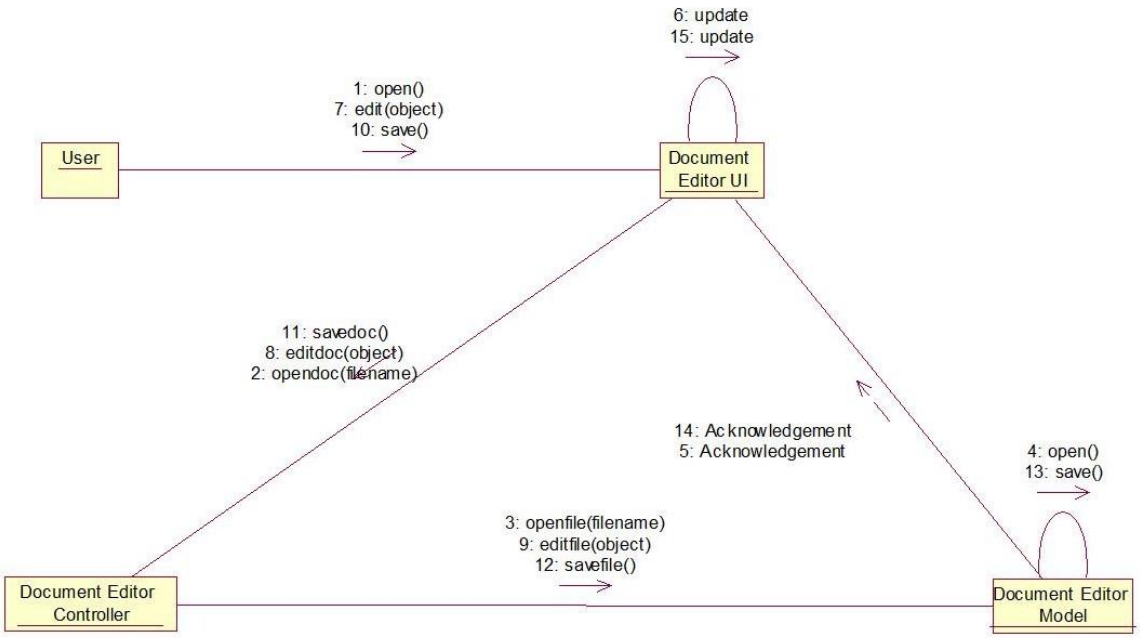

Fig: 3.5.1

CHAPTER 4

FORM DESIGN

SAMPLE SCREENS:

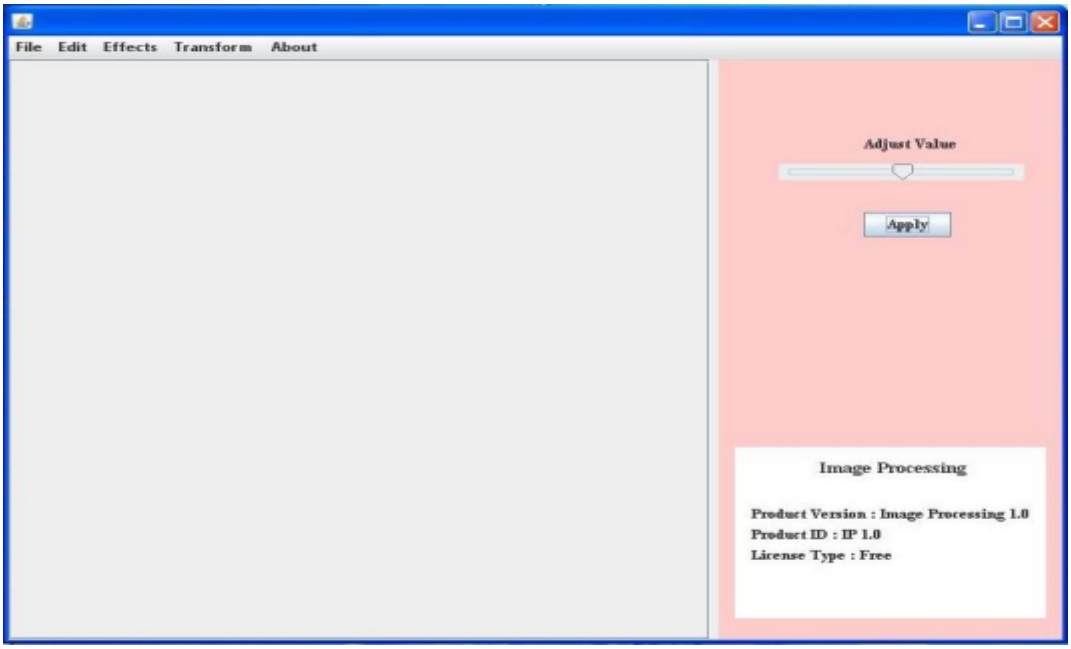

FIG 4.1: FILE MENU

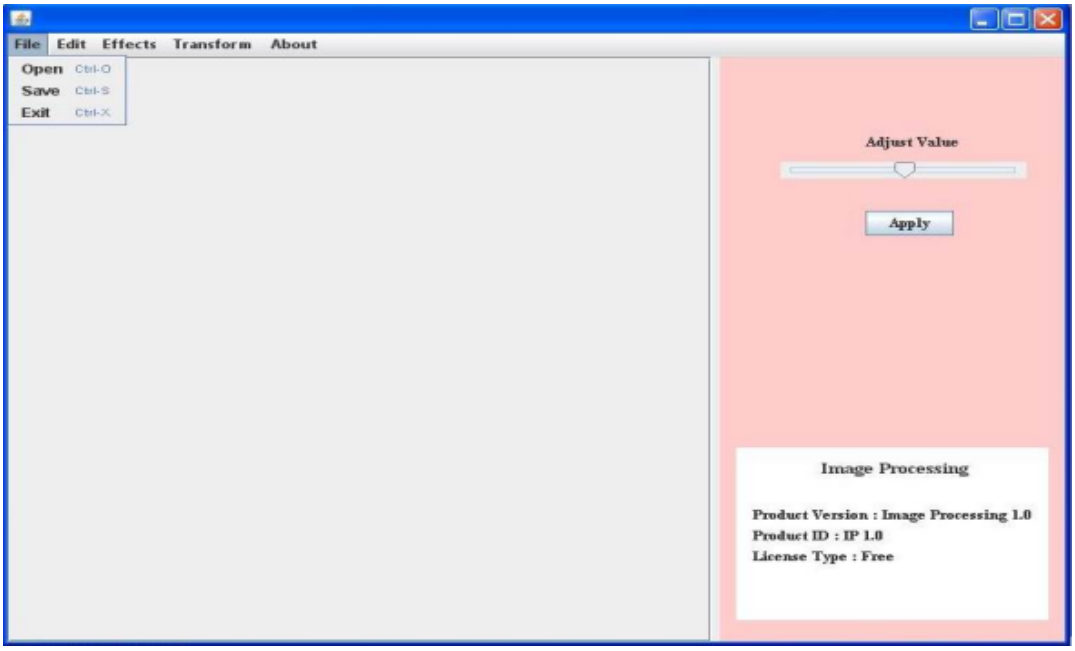

FIG 4.2: EDIT MENU

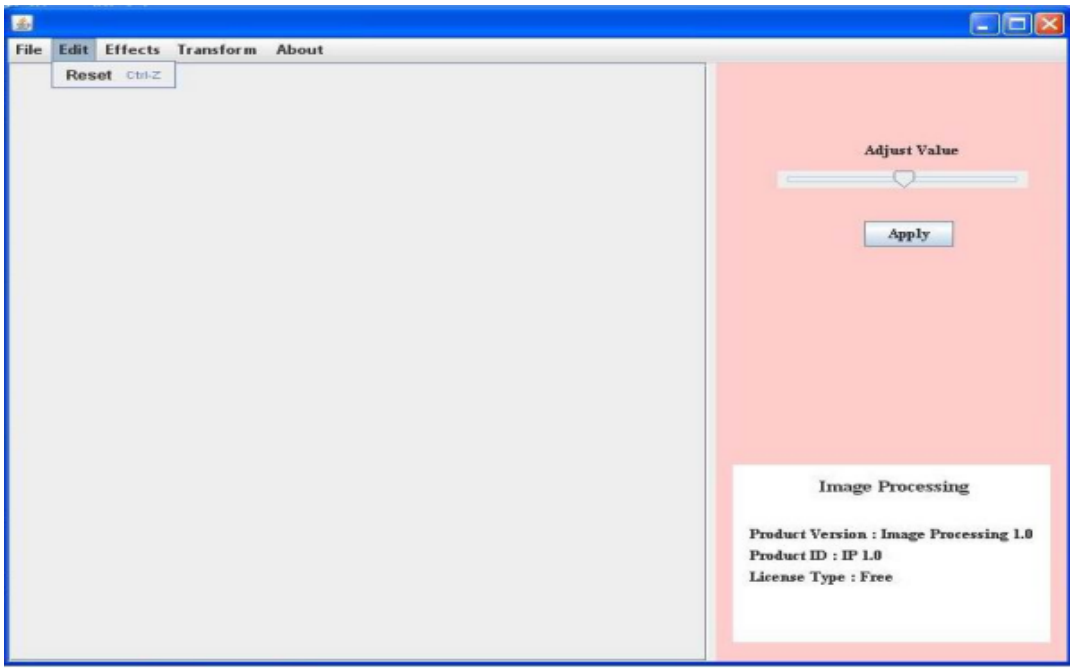

FIG 4.3: EFFECTS

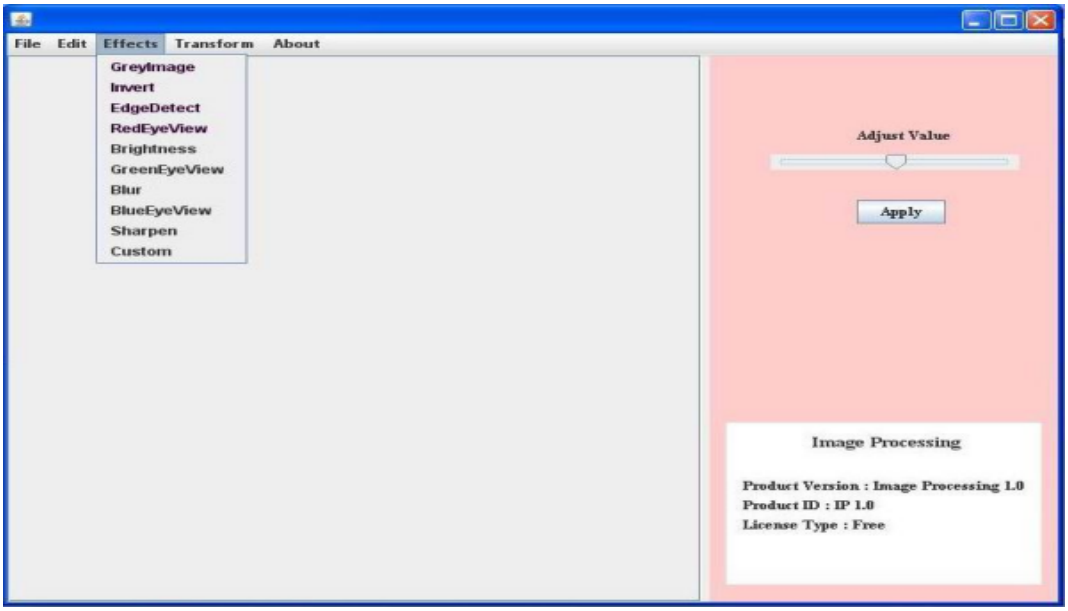

FIG 4.3.1: GREY IMAGE

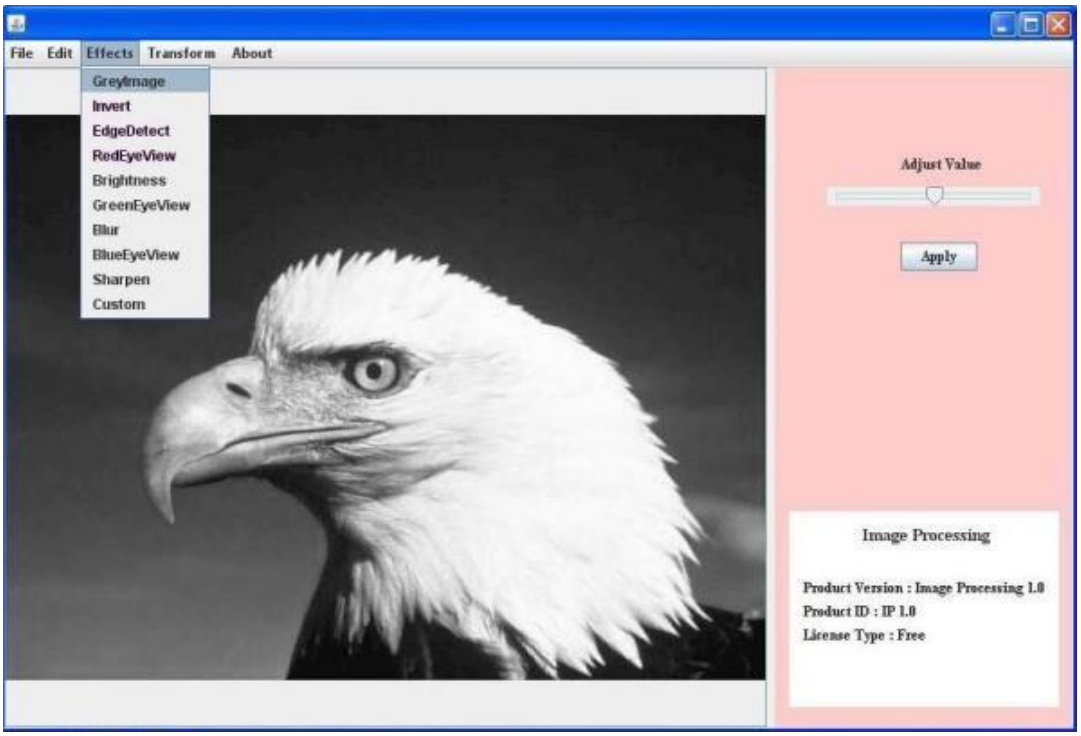

FIG 4.3.2: RED EYE

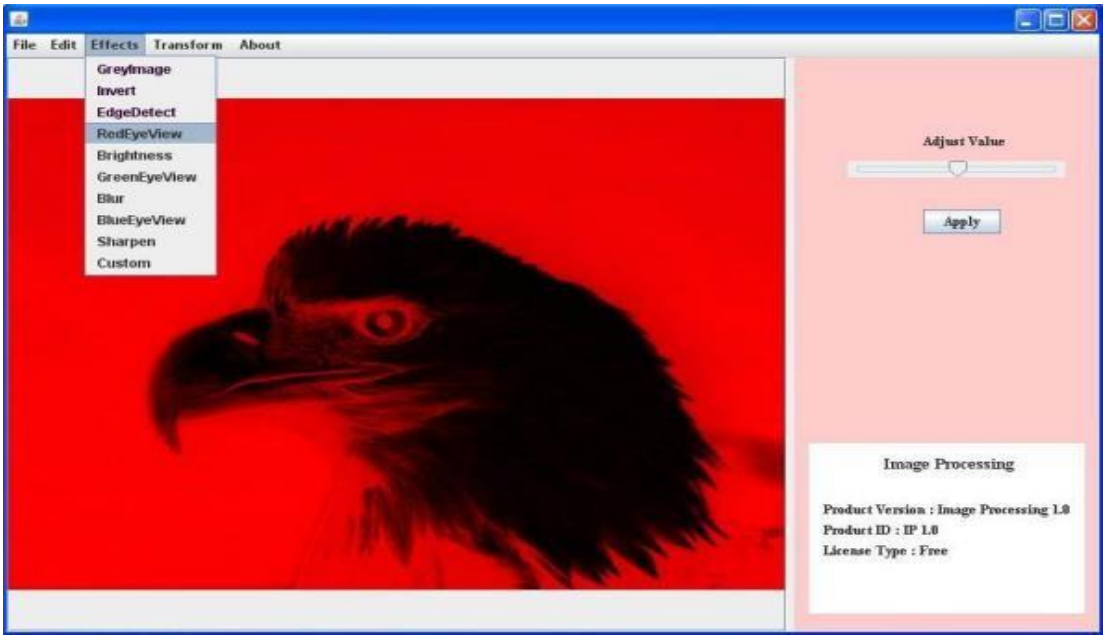

FIG 4.3.3: BRIGHTNESS

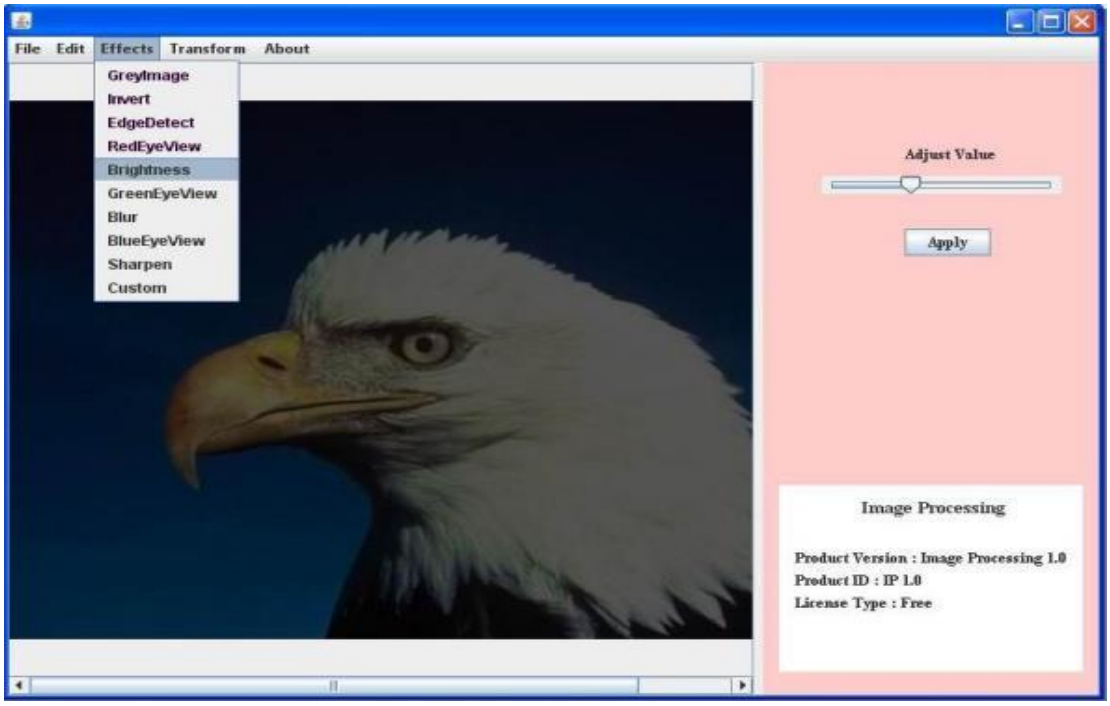

FIG 4.3.4: CUSTOM

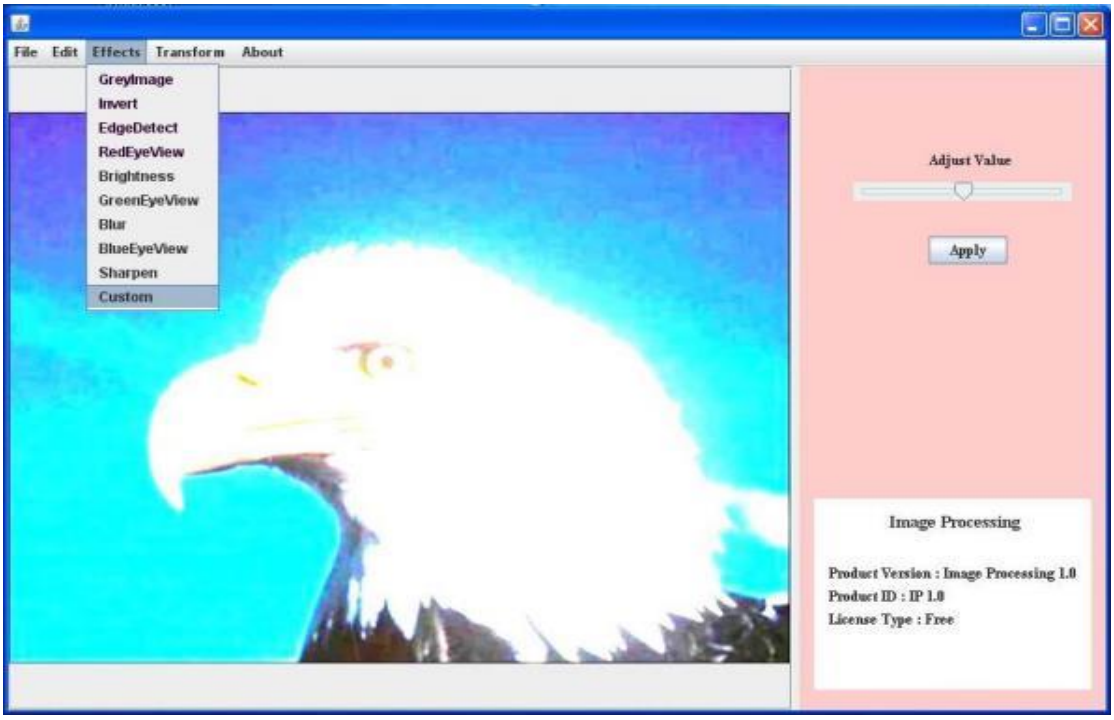

FIG 4.4: TRANSFORM MENU ITEMS

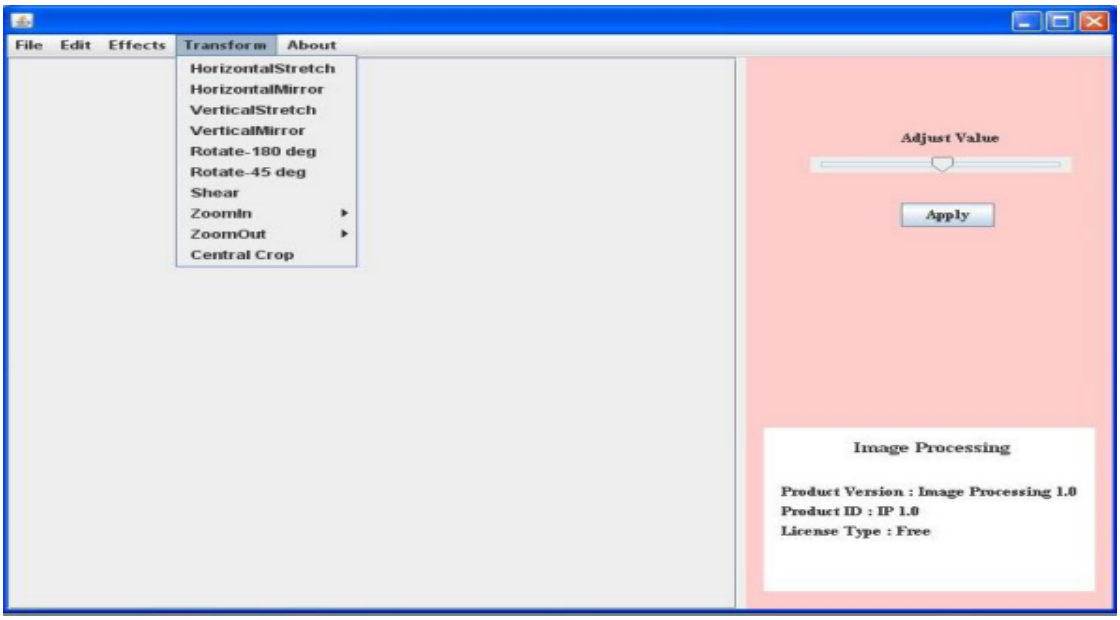

FIG 4.4.1: HORIZONTAL STRETCH

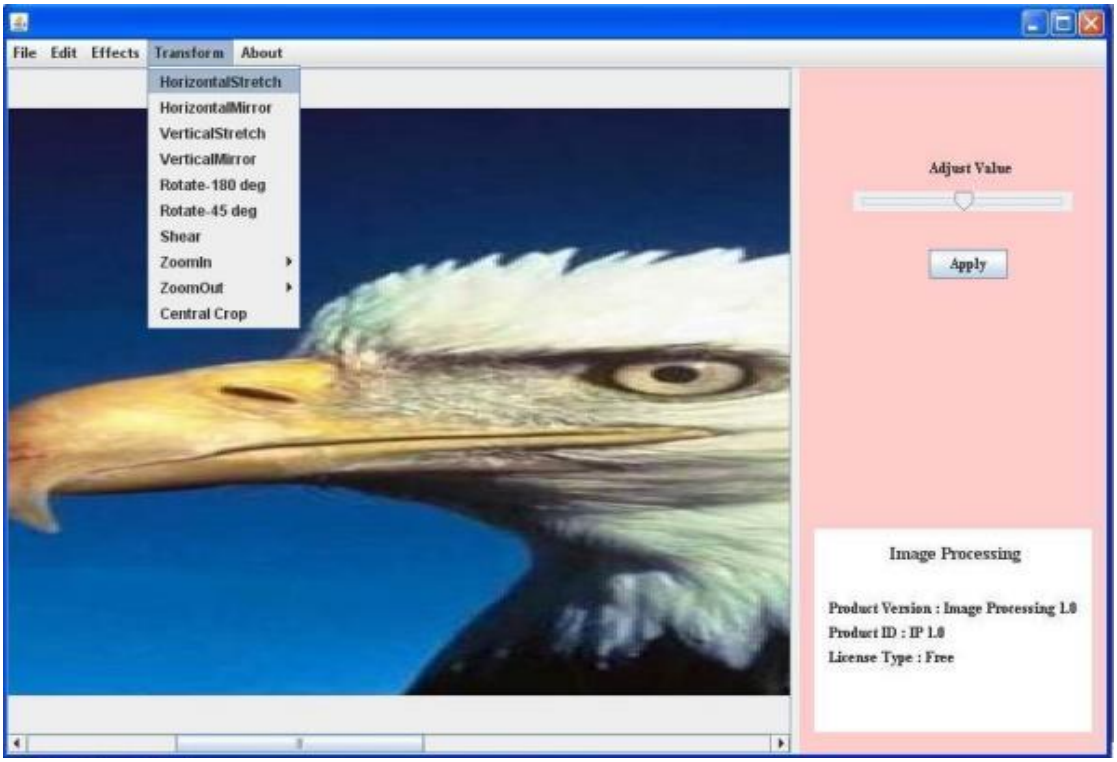

FIG 4.4.2: VERTICAL STRETCH

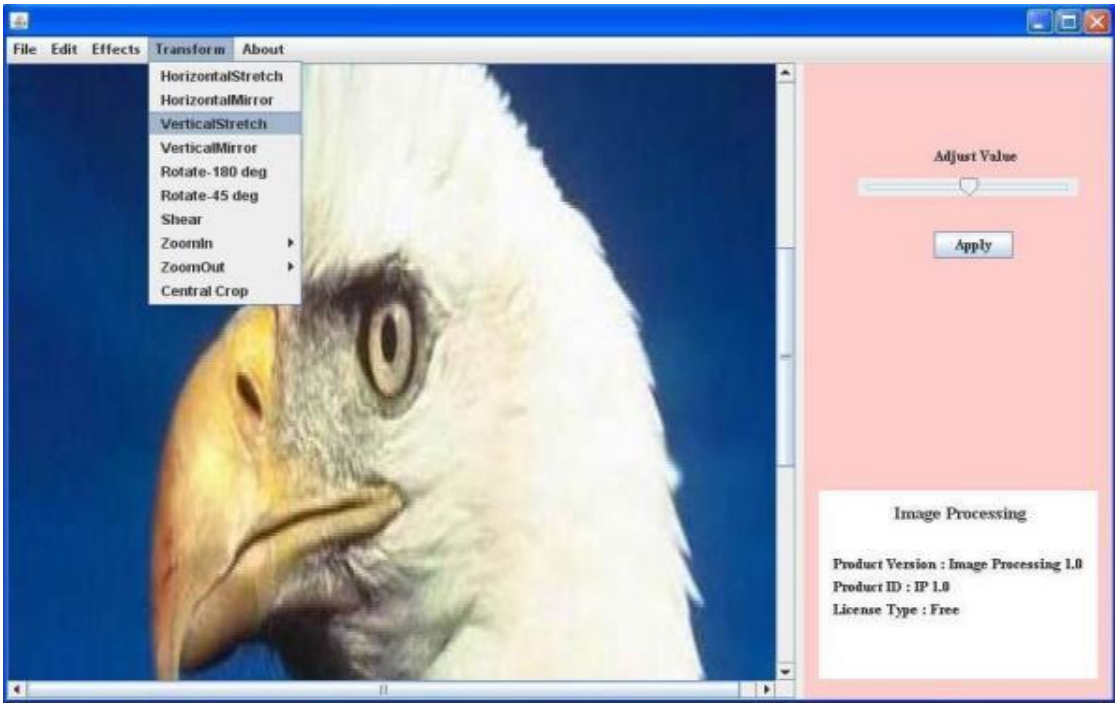

FIG 4.4.3: VERTICAL MIRROR

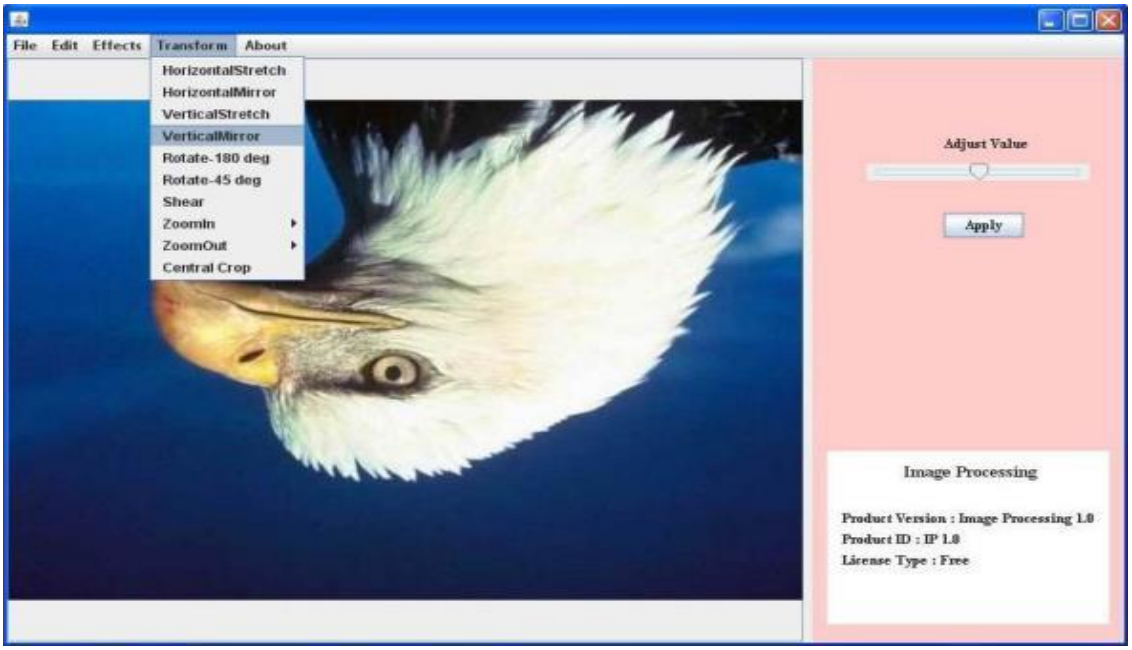

FIG 4.4.4: ROTATE-180 DEG

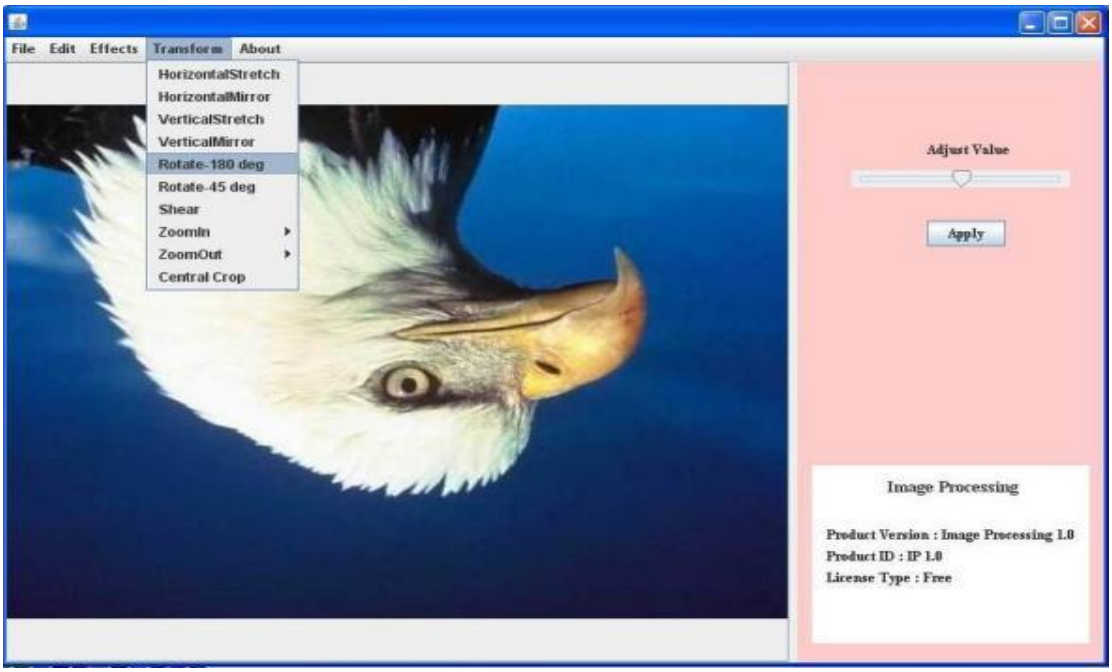

FIG 4.4.5: SHEAR

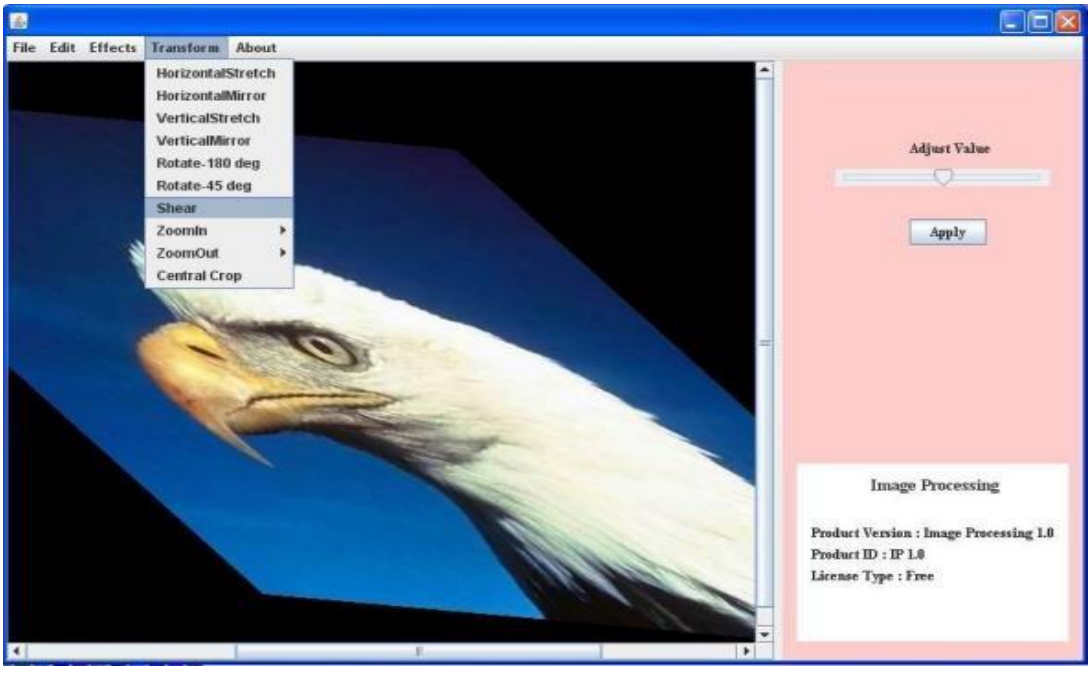

FIG 4.4.6: ZOOM IN

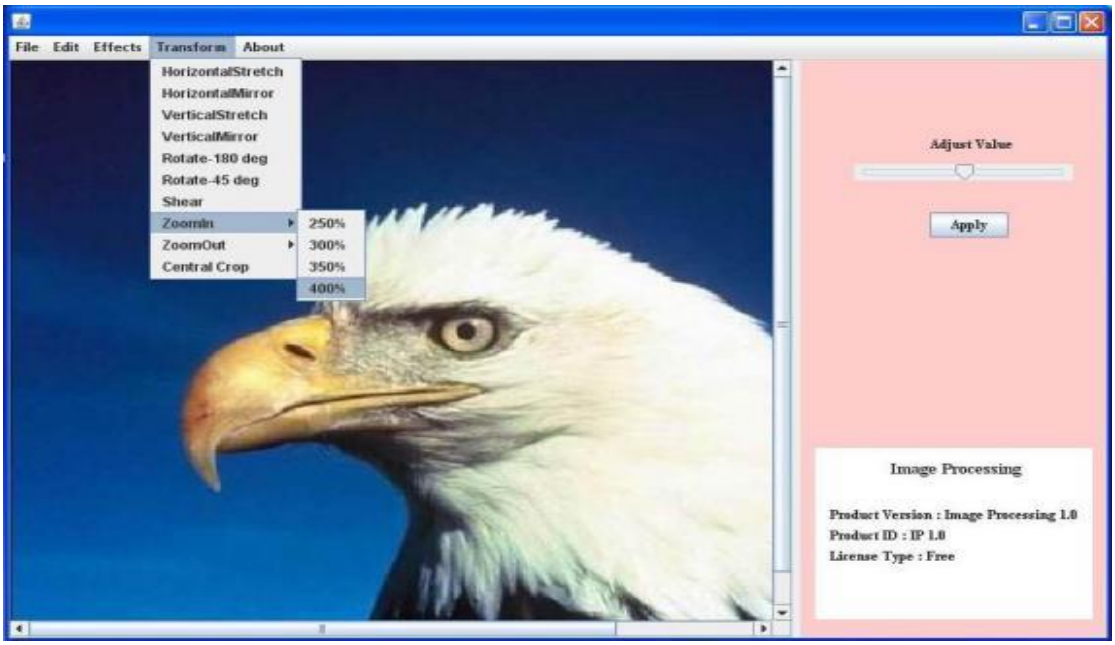

FIG 4.4.7: ZOOM OUT

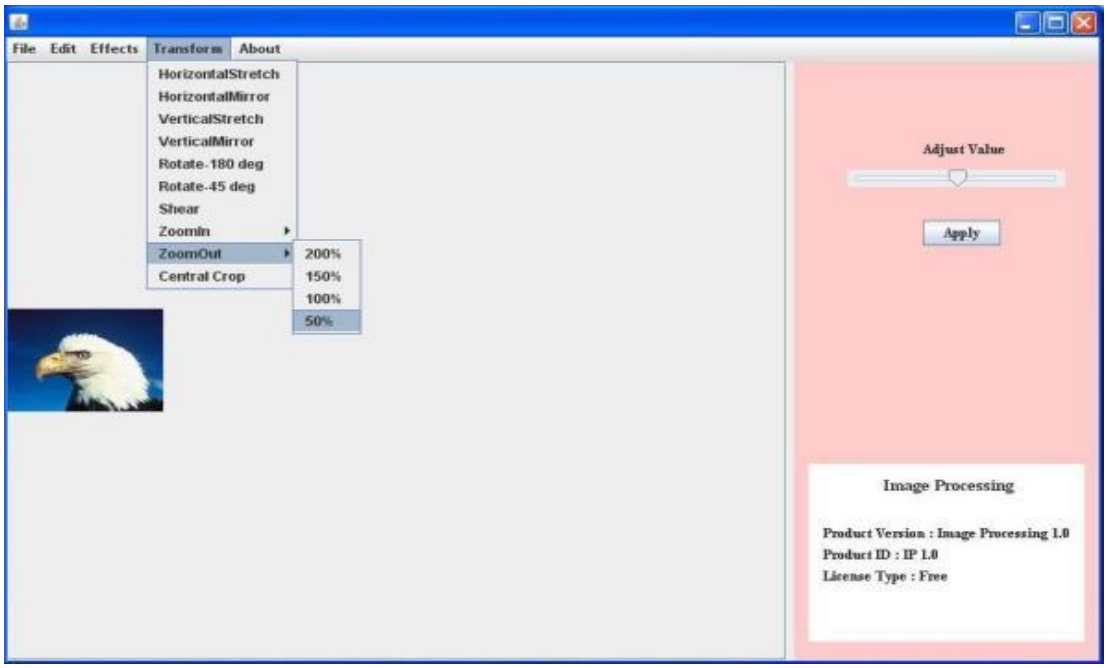

FIG 4.5: ABOUT MENU

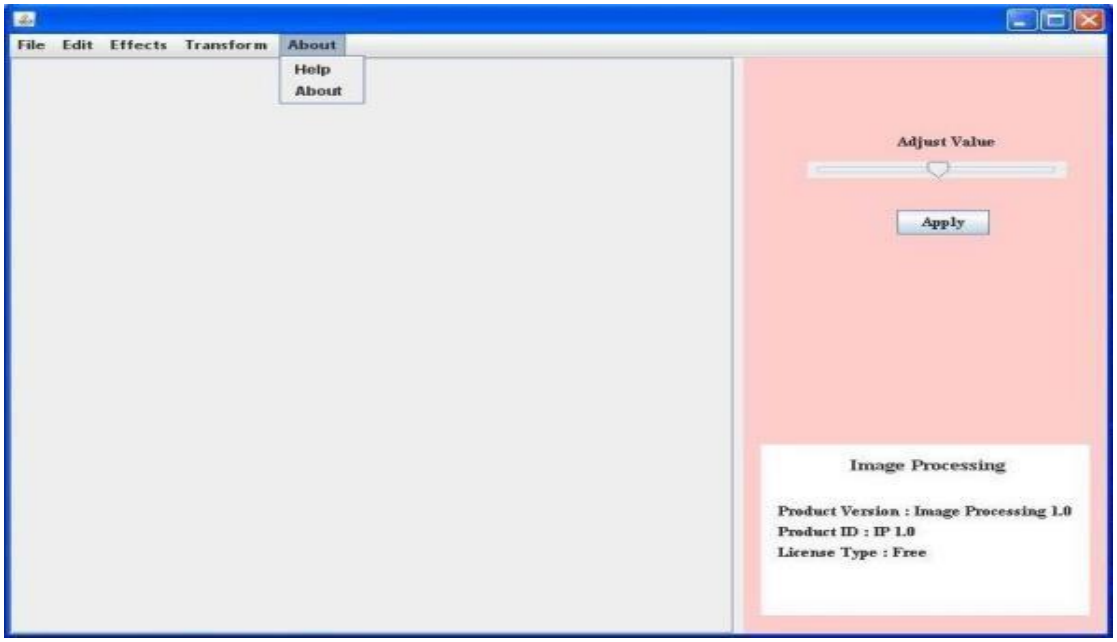

FIG 4.5.1: ABOUT:

CHAPTER 5

TESTING

5.1 UNIT TESTING

The first level of testing is unit testing. In this, the smallest units of software design, the module are tested against the specifications processed during the design for the modules are tested against the specification produced during the design for the modules. Unit testing is essentially for verification of the code produced during the coding phase and hence the goal is to test the internal logic of the modules. Unit testing involved checking all pages for errors and omissions. We used the unit testing plans prepared in the design phase of the system development as guide.

5.2 INTEGRATION TESTING:

Data can be lost across an interface, one module can have an adverse effect on the other sub functions, when combined may not produce the desired functions. Integrating testing is systematic testing to uncover the errors within the interface. This testing is done with simple data and the development system has run successfully this simple data. The need for an integrated system is to find the overall system performance.

5.3 SYSTEM TESTING:

In this testing, the entire software system is tested. All the application programs are grouped. together for the system testing, to test the whole system exhaustively including any additional housekeeping function like file achieving. This is the developers the last opportunity to check. that the system works before asking the silent to accept it. The purpose of this testing is to verify if the software meets its requirements, it is also tests to find discrepancies between the system and its original objective specification and system document. After this test, it was found that our project “IMAGE PROCESSING TECHNIQUES” works well as per the specified requirements.

5.4 ACCEPTANCE TESTING:

User Acceptance Testing of the system is the key factor for the success of any. system. The system under considerations is tested for the user acceptance by constantly keeping in touch with perspective system at the time of development and making change whenever required. This is done with regard to the input screen design.

BIBLIOGRAPHY

[1] Theo Pavlidis, Algorithms for Graphics and Image Processing, Computer Science Press, 1982.

[2] John C. Russ, the Image Processing Handbook, CRC Press, 1995.

[3] Ernest L. Hall, Computer Image Processing, Academic Press, 1979.

[4] David Bitsy [et al.], Complete Guide to MMX Technology, McGraw-Hill 1997.

[5] Joe King, MATLAB for Engineers – Second Edition, Addison-Wesley Publishing Co., 1998.

[6] Barry Haynes, Photoshop 6 Artistry – Mastering the Digital Image, New Riders Publishing, 2001.

[7] Lawrence H. Rodriguez, Building Imaging Applications with Java Technology, Addison-Wesley Publishing Co., 2001.

[8] Michael J. Laslo, Object-Oriented Programming featuring Graphical Applications in Java, Addison-Wesley Publishing Co., 2001.

[9] Mitchell Waite [et al.], Data Structures and Algorithms in Java, Waite Group Press, 1999.

[10] Matthew Robinson [et al.], Swing, Manning

[11] Daniel M Hoffman, Software Design, Automated Testing and Maintenance, International Thompson Computer Press, 1995.

[12] C. T. Arrington, Enterprise Java with UML, John Wiley and Sons, 2001.

[13] Image Processing in Java – Riyaz Mansoor

[14] www.intel.com

[15] www.microsoft.com

[16] www.imagemagick.org