VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI - 590018

Mini Project Report

On

"IMAGE CROPPER: SELECT AND SAVE A PORTION OF THE IMAGE"

A report submitted in partial fulfillment of the requirements for

Mini Project
In
BASIC DIGITAL IMAGE PROCESSING
Submitted by

DURGESH SHETTY	**4AL21AI012**
MOHAMMAD ZAAFIR	**4AL21AI024**
NITHEESH	**4AL21AI026**
PRAJWAL RAO	**4AL21AI031**

Under the Guidance of

Dr. Ganesh K

Senior Assistant Professor

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY MIJAR,
(Unit of Alva's Education Foundation ®, Moodbidri)
Affiliated to Visvesvaraya Technological University, Belagavi,
Approved by AICTE, New Delhi, Recognized by Government of Karnataka.
Accredited by NACC with A+ Grade
Shobavana Campus, Mijar, Moodbidri, D.K., Karnataka

2024 – 2025

ALVA'S INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Unit of Alva's Education Foundation ®, Moodbidri)
Affiliated to Visvesvaraya Technological University, Belagavi,
Approved by AICTE, New Delhi, Recognized by Government of Karnataka.
Accredited by NACC with A+ Grade
Shobavana Campus, Mijar, Moodbidri, D.K., Karnataka

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

CERTIFICATE

This is to certify that the Mini Project entitled **"IMAGE CROPPER: SELECT AND SAVE A PORTION OF THE IMAGE"** has been successfully completed and report submitted in A.Y 2024-25. It is certified that all corrections/suggestions indicated Presentation session have been incorporated in the report and deposited in the department library.

The assignment was evaluated and group members marks as indicated below

SI	USN	NAME	Presentation Skill (5)	Report (10)	Subject Knowledge (5)	Total Marks (20M)
1	4AL21AI012	DURGESH SHETTY				
2	4AL21AI024	MOHAMMAD ZAAFIR				
3	4AL21AI026	NITHEESH				
4	4AL21AI031	PRAJWAL RAO				

Dr. Ganesh K

Senior Assistant Professor

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany a successful completion of any task would be incomplete without the mention of the people who made it possible, success is the epitome of hardwork and perseverance, but steadfast of all is encouraging guidance.

So, with gratitude, we acknowledge all those whose guidance and encouragement served as a beacon of light and crowned the effort with success.

The selection of this mini-project work as well as the timely completion is mainly due to the interest and persuasion of our Project guide **Dr. Ganesh K,** Senior Assistant Professor, Department of Artificial Intelligence & Machine Learning. We will remember his contribution forever.

We sincerely thank, **Prof. Harish Kunder**, Associate Professor and Head of the Department of Artificial Intelligence & Machine Learning who has been the constant driving force behind the completion of the project.

We thank our beloved Principal, **Dr. Peter Fernandes,** for his constant help and support throughout.

We are indebted to the **Management of Alva's Institute of Engineering and Technology, Mijar, Moodbidri** for providing an environment which helped us in completing our mini project.

Also, we thank all the teaching and non-teaching staff of Department of Artificial Intelligence & Machine Learning for the help rendered.

DURGESH SHETTY	**4AL21AI012**
MOHAMMAD ZAAFIR	**4AL21AI024**
NITHEESH	**4AL21AI026**
PRAJWAL RAO	**4AL21AI031**

ABSTRACT

This project focuses on developing a basic image cropping system using Scilab, an open-source computational software, to perform essential image manipulation tasks such as reading, selecting a region of interest, cropping, and saving processed images. The system allows users to input an image file, interactively select a portion of the image for cropping to focus on specific areas of interest, and save the output as a new file. Cropping simplifies the image data by eliminating unnecessary regions, retaining only the relevant features. The user-friendly interface ensures smooth interaction, with prompts for file paths, image names, and intuitive region selection. Error handling prevents issues with missing, invalid files, or incorrect selections. Additionally, the system displays both the original and cropped images side by side using subplots, offering a clear comparison of the input and processed images. This functionality helps users visually understand the impact of image cropping. The modular design allows for future enhancements, such as incorporating advanced features like resizing, aspect ratio adjustment, and batch cropping. The project serves as an introductory tool for understanding basic image processing while providing a foundation for more complex applications in the future.

TABLE OF CONTENTS

CHAPTER NO.		DESCRIPTION	PAGE NO.
1		INTRODUCTION	1
	1.1	Brief Introduction	1
	1.2	Contribution Of The Work	1
2		RELATED WORKS	2
	2.1	Recent Year Paper	2
3		PROBLEM STATEMENT	3
	3.1	Problem Statement	3
	3.2	Objectives	3
4		SYSTEM ARCHITECTURE	4
	4.1	Core Logic/Main Functions	4
5		FEATURES AND FUNCTIONS	5
	5.1	Features And Functions	5
6		PROPOSED SYSTEM	6-7
	6.1	Architecture	6
	6.2	Algorithms Used	6
7		RESULTS AND SAMPLE CODE	8-9
	7.1	Results	8
	7.3	SAMPLE CODE	8
8		CONCLUSION AND FUTURE ENHANCEMENT	10
		REFERNECES	11

TABLE OF FIGURE

Fig No.	DESCRIPTION	PAGE NO.
7.1	OUTPUT	8
7.2	OUTPUT	8

CHAPTER 1

INTRODUCTION

1.1 BRIEF INTRODUCTION

This mini-project involves basic image processing using Scilab. The goal is to develop a program that can read an image, allow the selection of a region of interest, crop it, and save the processed image as a new file. The project introduces essential image handling functions such as imread() for reading images and imwrite() for saving them.

The project demonstrates cropping an image interactively, letting users focus on specific areas of interest. Cropping simplifies images by retaining only the relevant portion, making it valuable for preprocessing and analysis tasks.

By completing this project, you'll gain a foundational understanding of image manipulation in Scilab, laying the groundwork for more complex image processing tasks like resizing, segmentation, and object detection. The ability to work with images and apply basic transformations is crucial for advancing in the field of computer vision.

1.1 CONTRIBUTION OF THE WORK

The primary contribution of this work is the development of a basic image processing tool using Scilab, which allows users to perform fundamental operations such as reading, converting, and saving images in different formats. The tool is designed to be a simple and efficient solution for processing images and is aimed at users with basic to intermediate knowledge of image processing.

In particular, the project demonstrates the effective use of Scilab's image handling functions such as imread() for reading images from disk, rgb2gray() for converting a color image to grayscale, and imwrite() for saving the processed images. This provides an accessible platform for users to understand the core concepts of image processing while utilizing Scilab’s open-source environment.

Additionally, the system's ability to easily convert a color image to grayscale simplifies the process of image manipulation. Grayscale conversion is a crucial step in many image processing applications, as it reduces computational complexity while maintaining key features of the image. The system can be expanded in the future to include more advanced image processing techniques, such as noise reduction, edge detection, and object recognition, making it a stepping stone toward more complex computer vision applications.

CHAPTER 2

RELATED WORKS

2.1 RECENT YEAR PAPER

Digital image processing has seen extensive application across diverse domains, offering innovative solutions to complex challenges. Andreeva et al. (2024) introduced an interactive resource within the SCILAB platform, emphasizing its utility in image processing education. Their work highlights the potential of SCILAB as a tool for enhancing understanding and practical engagement in image processing tasks.

Velan and Poluru (2020) explored the application of digital image processing techniques in evaluating the quality of ARC and MIG welding of steel joints. Their study demonstrated the role of image processing in improving industrial quality control processes, particularly in assessing structural integrity and weld quality.

Bharodiya and Gonsai (2020) proposed an intelligent assistive algorithm leveraging binary Blob analysis for detecting bone tumors in X-ray images. This research exemplifies the integration of image processing techniques in medical diagnostics, providing a framework for efficient and accurate detection of abnormalities.

These studies collectively underline the versatility of digital image processing and its transformative impact across educational, industrial, and medical fields. Drawing inspiration from these advancements, our project focuses on developing a robust image cropping tool to enhance image processing workflows.

CHAPTER 3

PROBLEM STATEMENT

3.1 PROBLEM STATEMENT

In the digital era, image cropping has become an essential operation in fields like design, photography, research, and surveillance. However, the availability of cost-effective, user-friendly tools for precise cropping and saving selected portions of images is limited. Scilab, an open-source numerical computational software, offers great potential for these tasks but is often overlooked for image manipulation. The challenge lies in developing an intuitive yet efficient image cropping system that utilizes Scilab's capabilities, enabling users to focus on areas of interest while being accessible to both beginners and professionals.

3.2 OBJECTIVES

1. **To develop a basic image processing tool using Scilab** that enables users to perform essential operations such as reading images, converting them to grayscale, and saving them in different formats.
2. **To demonstrate the potential of Scilab for image manipulation** by leveraging its built-in functions and toolboxes for efficient and accurate processing.
3. **To provide an intuitive platform for learning and experimenting** with fundamental image processing concepts, suitable for users at various skill levels.
4. **To explore the application of grayscale conversion in image processing,** showcasing its importance in simplifying image analysis while preserving critical information.
5. **To lay the groundwork for expanding the system** with advanced image processing features, such as filtering, edge detection, and object recognition, in future iterations.

CHAPTER 4

SYSTEM ARCHITECTURE

4.1 CORE LOGIC/MAIN FUNCTIONS

The system architecture for this project consists of three primary components: Input Module, Processing Module, and Output Module, all seamlessly integrated using Scilab's built-in image processing functions. Below is an in-depth overview of the architecture and the core logic that powers the image cropping system.

- **INPUT MODULE**

The Input Module handles the process of receiving the image file from the user. It employs the imread() function to load the specified image. Before moving forward, the system checks the existence and format compatibility of the input file. If the file is missing or incompatible, the module generates an error message, ensuring that only valid files are accepted for cropping.

- **PROCESSING MODULE**

The Processing Module manages the core image manipulation tasks, including cropping the selected region. Using interactive tools, the system allows users to define a region of interest, isolating the desired portion of the image. This module emphasizes the importance of precise selection in focusing on relevant details and prepares the cropped image for saving or further use.

- **OUTPUT MODULE**

The Output Module oversees saving the cropped image. It utilizes the imwrite() function to save the selected portion of the image to a user-specified location and in the desired file format. Upon successful completion, the system displays a confirmation message to notify the user that the image has been saved correctly. This module ensures seamless output file management and enhances user satisfaction.

- **CORE LOGIC**

The Core Logic revolves around three primary functions: imread(), rgb2gray(), and imwrite(). These functions facilitate reading, processing, and saving images, forming the backbone of the system. Additional helper functions like disp() are used for displaying messages and ensuring a seamless user experience. This architecture is modular and can be extended to include advanced image processing techniques in the future.

CHAPTER 5

FEATURES AND FUNCTIONS

5.1 FEATURES AND FUNCTIONS

The developed image cropping system is designed with several features to ensure a smooth and efficient user experience. A key feature is its intuitive interface, which allows users to easily select the portion of the image they wish to crop and specify the output file name. The system provides clear feedback through error or success messages at each step, helping users navigate the process and ensuring seamless functionality.

- **CROPPING AND SAVING IMAGE SECTION**

Another key feature is image cropping, which allows users to select and isolate a specific portion of the image. This process simplifies the image by focusing on relevant details while removing unnecessary areas. Cropping is essential in many practical applications, such as object recognition, feature extraction, and image analysis, where attention is needed only on specific parts of the image.

- **FILE HANDLING CAPABILITIES**

The system incorporates strong file handling features to ensure compatibility with popular image formats like .jpg and .png. It includes error handling mechanisms that identify invalid or missing files, preventing crashes or interruptions. This functionality makes the tool dependable and easy to use, especially for beginners, by ensuring a smooth image cropping and saving process.

- **SCILAB**

An additional advantage is that the system is built on **Scilab**, an open-source computational software. This ensures accessibility for a wide audience without the need for costly licenses. The use of open-source tools makes the system an ideal solution for educational and experimental purposes.

CHAPTER 6

PROPOSED SYSTEM

6.1 ARCHITECTURE

- **LIBRARIES USED:**

SCILAB: Scilab serves as the primary platform for image processing in this project. It is an open-source software used for numerical computation, offering a wide range of tools for signal processing, image manipulation, and more.

IMAGE_PROCESSING_TOOLBOX: This toolbox includes functions such as imread, imshow, and imwrite, which allow for reading, displaying, and saving images.

MATPLOTLIB: Although Scilab lacks a direct plotting library like Python, matplotlib can be utilized for advanced visualization if necessary.

OS: This module provides functions for interacting with the operating system, including file and directory management.

- **Description of Libraries:**

SCILAB: The central computational platform used for image manipulation, including matrix operations and image transformations.

IMAGE_PROCESSING_TOOLBOX: A toolkit for basic image processing operations, enabling the user to read, process, and save images in various formats.

MATPLOTLIB: A library for creating advanced visualizations and subplots, helping in comparing processed images (if required).

OS: Used for handling file paths, directories, and managing input/output files in the project.

6.2 ALGORITHMS USED

ALGORITHMS USED FOR IMAGE PROCESSING:

Cropping And Saving Section:

Cropping is the process of selecting and isolating a specific portion of an image. In this project, Scilab's interactive tools are used to define and crop the desired region. This approach simplifies image data by focusing on relevant sections while preserving the essential structure and details within the selected area.

Input Image: The image is loaded into a matrix using the imread() function.

Cropping the Image: The user selects a region of interest interactively, isolating the desired portion of the image for cropping.

Image Saving: The cropped image is saved using the imwrite() function. It is stored at the specified output location in formats like .jpg, .png, etc.

Subplot Visualization: The system displays the original and cropped images side by side using subplot(). This provides an easy comparison between the full image and the selected portion, enhancing visualization.

Display Original Image: The full, unmodified image is shown in one subplot.

Display Cropped Image: The cropped section of the image is displayed in the second subplot.

CHAPTER 7

RESULTS AND SAMPLE CODE

7.1 RESULTS

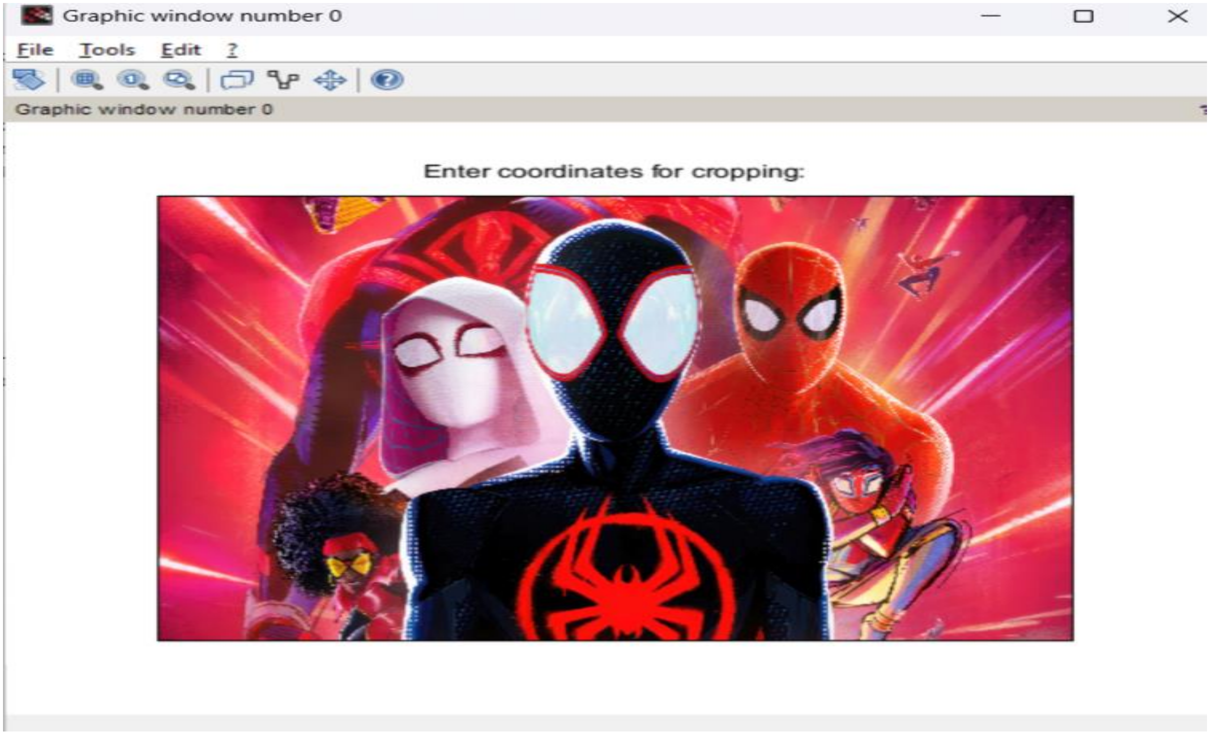

Figure 7.1 Output

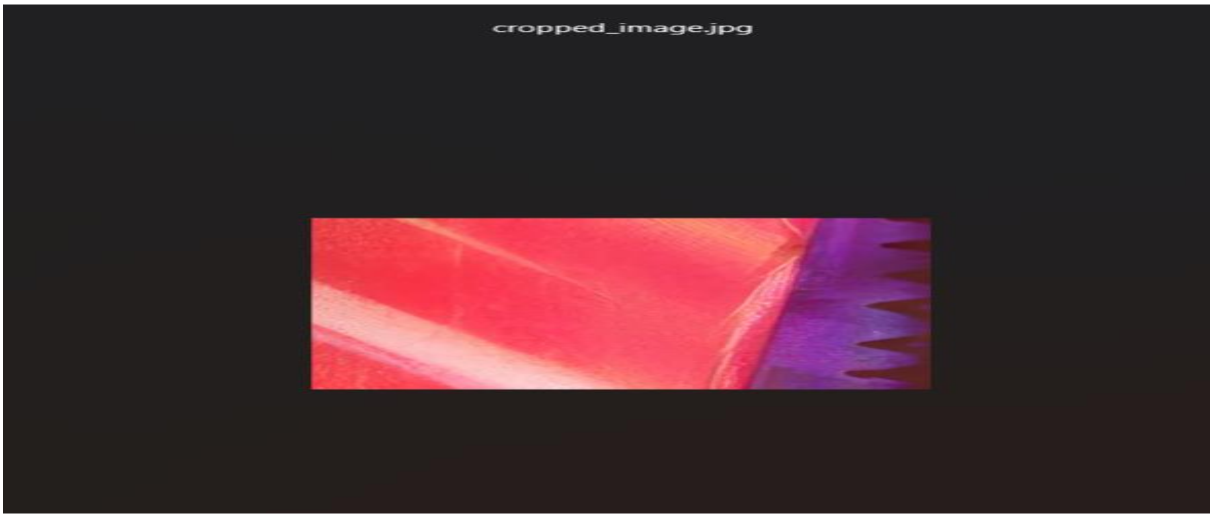

Figure 7.2 Output

7.3 SAMPLE CODE

```
// Load the image

img = imread("C:/sciii/spiderman.jpg");

// Display the image

imshow(img);

title('Enter coordinates for cropping:');

// Prompt the user for the top-left and bottom-right coordinates

x1 = input('Enter the x-coordinate for the top-left corner: ');

y1 = input('Enter the y-coordinate for the top-left corner: ');

x2 = input('Enter the x-coordinate for the bottom-right corner: ');

y2 = input('Enter the y-coordinate for the bottom-right corner: ');


// Ensure the coordinates are valid

if x1 > x2 then

    temp = x1;

    x1 = x2;

    x2 = temp;

end

if y1 > y2 then

    temp = y1;

    y1 = y2;

    y2 = temp;

end

// Crop the image using the selected coordinates
```

```
cropped_img = img(y1:y2, x1:x2, :);

// Display the cropped image

imshow(cropped_img);

title('Cropped Image');

// Save the cropped image to a file

imwrite(cropped_img, 'cropped_image.jpg');
```

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENT

In conclusion, this project successfully implemented a basic image cropping system using Scilab, focusing on essential tasks such as reading an image, selecting a specific region, cropping it, and saving the processed image. The system also provides a side-by-side visualization of the original and cropped images using subplots, offering a clear comparison and enhancing the user's understanding of the transformation. Leveraging Scilab's robust image processing functions like imread and imwrite, the project demonstrates how core image manipulation tasks can be effectively performed in an open-source scientific computing environment. This implementation serves as a strong starting point for more advanced image processing capabilities.

Looking ahead, several enhancements could be introduced to improve functionality and user experience. Future advancements could include the addition of tools for precise cropping, such as adjustable selection boxes or coordinate-based cropping. Developing an interactive graphical user interface (GUI) would simplify user interaction, making the system more accessible to beginners. Extending support for batch processing would allow users to crop and save multiple images simultaneously, increasing efficiency. Additionally, integrating features such as automated object detection for cropping and incorporating machine learning models for advanced image analysis could broaden the system's applicability in fields like design, research, and surveillance.

REFERENCES

[1] Scilab Consortium. (2021). *Scilab: A Free and Open Source Software for Engineering and Scientific Applications.* Retrieved from https://www.scilab.org

Andreeva, H., Gabrova, R., Kuzmanov, I., & Technical Faculty, University of Food Technologies, Maritza blv. 26, 4000 Plovdiv, Bulgaria. (2021). INTERACTIVE RESOURCE FOR IMAGE PROCESSING IN SCILAB. In *Journal of Hygienic Engineering and Design* (pp. 16–17) [Review paper]. https://keypublishing.org/jhed/wp-content/uploads/2024/03/03.-Full-paper-Hristina-Andreeva.pdf

[3] Velan, S. S., & Poluru, V. R. (2020). Application of digital image processing techniques in determining the quality of ARC and MIG welding of steel joints. *2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 1320–1325. https://doi.org/10.1109/icrito48877.2020.9198021

[4] Bharodiya, A. K., & Gonsai, A. M. (2020). An intelligent assistive algorithm for bone tumor detection from human X-Ray images based on binary Blob analysis. *International Journal of Information Technology*, *14*(3), 1467–1473. https://doi.org/10.1007/s41870-020-00539-0