

**VIETNAM GENERAL CONFEDERATION OF LABOUR
TON DUC THANG UNIVERSITY
FACULTY OF INFORMATION TECHNOLOGY**



Nguyễn Đình Việt Hoàng – 522H0120

Nguyễn Đỗ Gia Hòa – 520H0633

**MIDTERM REPORT
INTRODUCTION TO DIGITAL IMAGE
PROCESSING**

HO CHI MINH CITY, YEAR 2024

**VIETNAM GENERAL CONFEDERATION OF LABOUR
TON DUC THANG UNIVERSITY
FACULTY OF INFORMATION TECHNOLOGY**



**Nguyễn Đình Việt Hoàng – 522H0120
Nguyễn Đỗ Gia Hòa – 520H0633**

**MIDTERM REPORT
INTRODUCTION TO DIGITAL IMAGE
PROCESSING**

HO CHI MINH CITY, YEAR 2024

ACKNOWLEDGEMENT

We sincerely thank Professor Trịnh Hùng Cường for teaching us the Introduction to Digital Image Processsing course with great enthusiasm. We want to express our deep appreciation for the dedication and professional knowledge that you shared with us. Through your classes, we gained a better understanding of the fundamental aspects of the Introduction to Digital Image Processing, thanks to your detailed explanations and practical applications. You helped us grasp the knowledge and apply it effectively. Finally, we extend our heartfelt gratitude to Professor Trịnh Hùng Cường for your commitment and invaluable support throughout our learning journey in this course. The skills and knowledge we acquired will continue to impact our future development. We sincerely thank you and wish you health, success, and happiness.

Ho Chi Minh City, April 9, 2024

Authors

(sign and write your full name)

Nguyễn Đỗ Gia Hòa

Nguyễn Đình Việt Hoàng

THE COMPLETION REPORT HAS BEEN SUBMITTED AT TON DUC THANG UNIVERSITY

Our group assures that this is our own report and was guided by Professor Trịnh Hùng Cường. The research content and results in this report are honest and have not been published in any form before. The figures in the tables used for analysis, comments, and evaluations were collected by the authors from various sources clearly stated in the reference section.

Additionally, the report includes some comments, evaluations, and data from other authors and organizations, all of which are cited and noted for their origin.

If any fraud is detected, we fully take responsibility for the content of our midterm report for the second semester of the 2023-2024 academic year. Ton Duc Thang University is not involved in any copyright or intellectual property violations that we may cause during the process (if any).

Ho Chi Minh City, April 9, 2024

Authors

(sign and write your full name)

Nguyễn Đỗ Gia Hòa

Nguyễn Đình Việt Hoàng

INSTRUCTOR RUBRIC

Supervisor's Name:

Comments:

Total Score Based on Rubric Evaluation:

Ho Chi Minh City, date ... month ... year ...

Supervisor

(sign and write your full name)

TABLE OF CONTENTS

CHAPTER 1. METHODOLOGY OF SOLVING TASKS	8
CHAPTER 2: TASK RESULTS	9

IMAGE CATALOG

Figure 1.1. The yellow star is extracted from the input image (input1.jpg).	9
Figure 1.2. The orange star is extracted from the input image (input1.jpg).	10
Figure 1.3. The pink star is extracted from the input image (input1.jpg).	11
Figure 1.4. The blue star is extracted from the input image (input1.jpg).	12
Figure 1.5. The green star is extracted from the input image (input1.jpg).	13
Figure 1.6. The purple star is extracted from the input image (input1.jpg).	14
Figure 1.7. The black stars is converted from the input image (input1.jpg).	15
Figure 2. Each digit is drawn rectangles surrounding automatically and the figure below is the output image of the input image (input2.png).	16

CHAPTER 1: METHODOLOGY OF SOLVING TASKS

II/ Programming part:

Programming Task 1:

1a and 1b:

1. Load the image: The image is loaded using the `cv2.imread()` function. This is done twice, once for each part of the code.
2. Define color ranges: For the first part of the code, color ranges for different colors (yellow, orange, pink, blue, green, purple) are defined using NumPy arrays.
3. Color extraction: A function `extract_color()` is defined to extract a specific color from the image. This is done by converting the image to HSV color space, creating a mask for the specified color range, applying the mask to the image, and reducing noise using morphological closing.
4. Create masks for specific colors: Masks for yellow and orange colors are created and combined. The combined mask is then inverted and morphological opening is applied to reduce noise.
5. Thresholding: The image is converted to grayscale, blurred using median blur, and then binary thresholding is applied.
6. Bitwise AND operation: The result of thresholding and the mask after morphological opening are combined using a bitwise AND operation.
7. Write images: The images after color extraction and the result of the bitwise AND operation are written to disk.

Programming Task 2:

1. For the second part, the image is loaded using the `cv2.imread()` function. This is done twice, once for each part of the code.
2. Convert to grayscale: The image is converted to grayscale using `cv2.cvtColor()`.
3. Blur the image: The grayscale image is blurred using Gaussian blur.
4. Adaptive thresholding: Adaptive thresholding is applied to the blurred image.
5. Morphological operations: Morphological closing and opening are applied to the thresholded image to remove noise and small holes.
6. Find contours: Contours are found in the image using `cv2.findContours()`.
7. Draw bounding boxes: For each contour, if it passes certain conditions (contour area and aspect ratio), a bounding box is drawn around it using `cv2.rectangle()`.
8. Write the image: The image with bounding boxes is written to disk.

CHAPTER 2: TASK RESULTS

Programming Task 1:

1a and 1b:

Figure 1.1. The yellow star is extracted from the input image (input1.jpg).

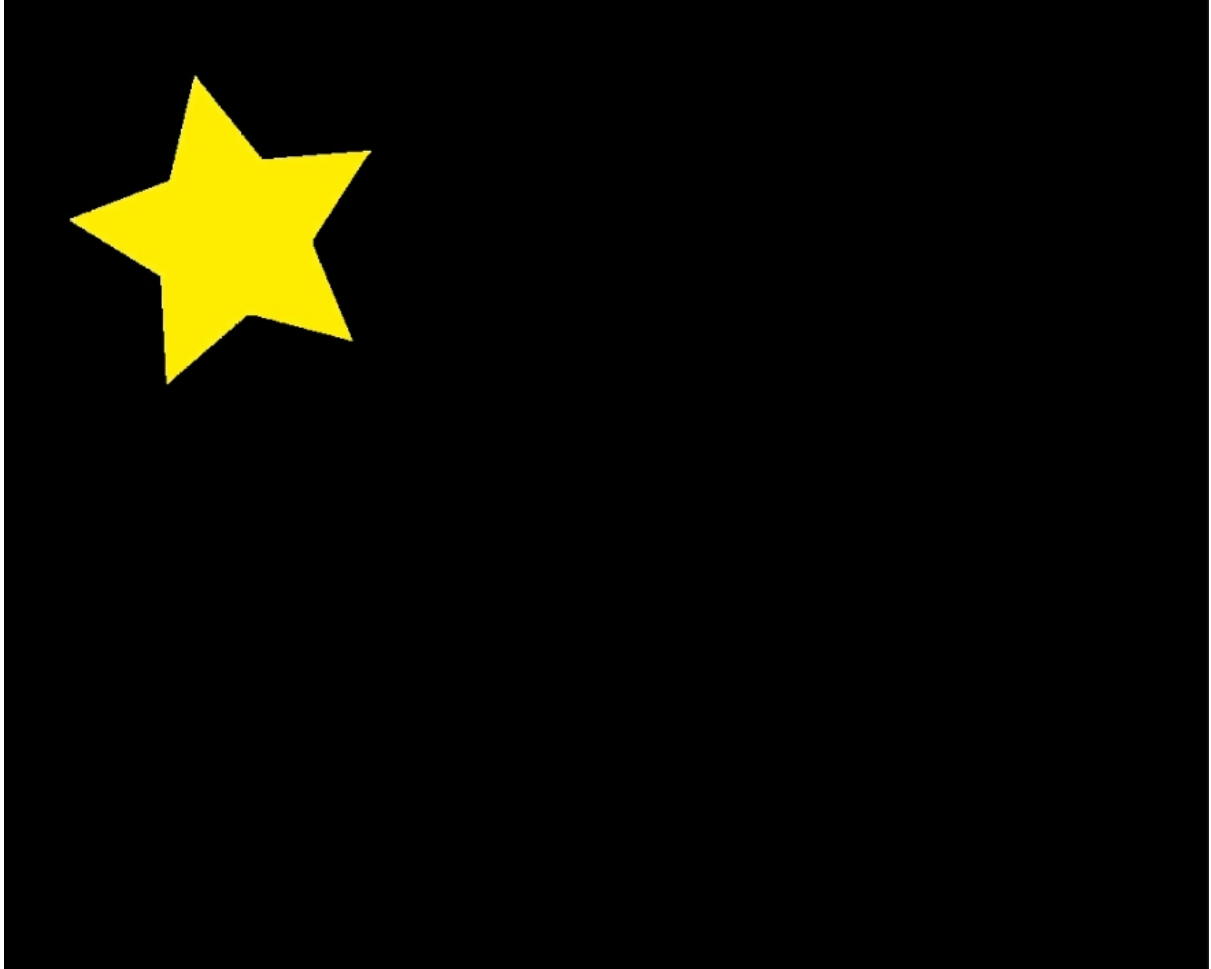


Figure 1.2. The orange star is extracted from the input image (input1.jpg).



Figure 1.3. The pink star is extracted from the input image (input1.jpg).



Figure 1.4. The blue star is extracted from the input image (input1.jpg).



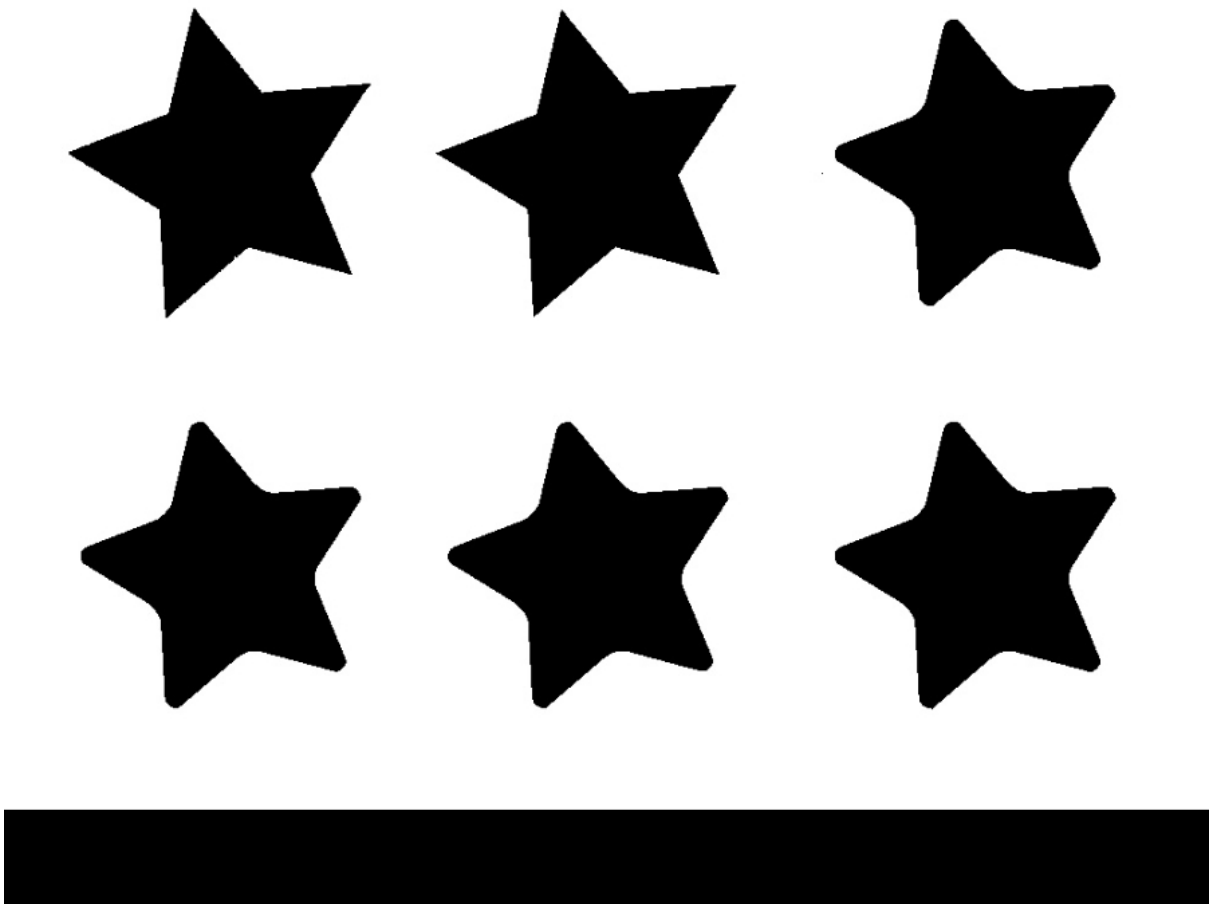
Figure 1.5. The green star is extracted from the input image (input1.jpg).



Figure 1.6. The purple star is extracted from the input image (input1.jpg).



Figure 1.7. The black stars is converted from the input image (input1.jpg).



Programming Task 2:

Figure 2. Each digit is drawn rectangles surrounding automatically and the figure below is the output image of the input image (input2.png).

