Vietnam General Confederation of Labor

TON DUC THANG UNIVERSITY

FACULTY OF INFORMATION TECHNOLOGY

MIDTERM ESSAY INTRODUCE TO DIGITAL IMAGE PROCESSING

Instructor: **Ph D. TRINH HUNG CUONG**

Student: **VU BAO AN – 521H0435**

Class : **21H50203**

Year : **2023-2024**

HO CHI MINH CITY, 2024

Vietnam General Confederation of Labor

TON DUC THANG UNIVERSITY

FACULTY OF INFORMATION TECHNOLOGY

MIDTERM ESSAY INTRODUCE TO DIGITAL IMAGE PROCESSING

Instructor: **Ph D. TRINH HUNG CUONG**

Student: **VU BAO AN – 521H0435**

Class : **21H50203**

Year : **2023-2024**

HO CHI MINH CITY, 2024

LETTER OF APPRECIATION

Dear Ton Duc Thang University and Ph D. Trinh Hung Cuong.

I wanted to express our heartfelt gratitude for the incredible opportunity to study and learn from your institution and your team. The knowledge and skills I have gained during our time at Ton Duc Thang University will undoubtedly shape my futures in meaningful ways.

Your dedication to providing exceptional education and resources to your students is truly inspiring, and I feel honored to have been a part of your community. Thank you for your unwavering support and encouragement throughout my studies, and for helping me to achieve our academic goals.

We would also like to extend a special thank you to Ph D. Trinh Hung Cuong for his guidance and mentorship. His expertise and passion for teaching have been invaluable to my growth and development, and I am grateful for his contributions to our education.

Once again, thank you for everything. I will cherish the experiences and knowledge gained at Ton Duc Thang University for years to come.

THIS PROJECT WAS COMPLETED AT TON DUC THANG UNIVERSITY

I fully declare this to be my own work. The content in this topic is honest and has not been published in any form before. The data in the tables of analysis, comments and reviews are collected by the author himself from various sources, as indicated in the references.

In addition, the project also uses a number of comments, assessments as well as data from other authors, other agencies and organizations, with citations and source annotations.

Should any frauds be found, I will take full responsibility for the content of my report. Ton Duc Thang University is not related to copyright and copyright violations caused by me during the implementation process (if any).

Ho Chi Minh city, 8th April, 2024

An

Vu Bao An

CONFIRMATION AND ASSESSMENT SECTION

Instructor confirmation section

Ho Chi Minh April, 2024

(Sign and write full name)

Evaluation section for grading instructor

Ho Chi Minh, April 2024

SUMMARY

This is a report of the Software Engineering subject of the Faculty of Information Technology Ton Duc Thang University

INDEX

CHAPTER 1: METHODOLOGY OF SOLVING TASKS

- In OpenCV library, I used **imread(image_name)** function to read an image.
- To change the color channel, I used the cvtColor function with the syntax **cvtColor(image_name, desired_color_channel)**. For example, if I want to change the color channel of the *input.jpg* image (represented by the variable *img*) to the HSV color space, the syntax would be:
  cv2.cvtColor(img, cv.COLOR_BGR2HSV)
  Similarly, if I want to convert an image to grayscale, the syntax would be: ***cv2.cvtColor(img, cv.COLOR_BGR2GRAY)***

1.1 Extract each star in the input image automatically bu using color filtering in HSV color space

To complete this task,

1. Convert the color channels of the image from RGB (BGR in Python) to HSV using the **cvtColor** function.
2. Defined the range of the X color in HSV.
3. Thresholded the HSV image (mask) to isolate the stars with the desired color (X) using the **inRange** function.
4. Used the **bitwise_and** function to combine the input image and the mask, resulting in the desired output.

1.2 Repaint White borders of all stars to Black color by using thresholding techniques.

- Convert the color channels of the image from RGB to grayscale using the cvtColor function.

- Apply different thresholding methods to set the pixels greater than 218 or 220 to be white (255), otherwise black (0).
- The edges of the stars are colored black using the numpy.where() statement with the condition that pixels have a value of 255.

1.3 Repaint the background to White color and repaint all stars to Blackcolor by using thresholding techniques.

Method 1:

- Creating a copy of the input image (img = img1.copy()) and converting the image to the Grayscale color space using the OpenCV library.
- A binary thresholding technique is applied to create a binary mask (thresh_06) by setting a threshold value of 200. This mask is used to identify stars in the image.
- The contours are found using the cv.findContours function with specific parameters.
- Repaints the entire background of the image to white (img[:] = [255, 255, 255]).
- The identified stars are repainted to black by drawing filled contours on the image (cv.drawContours).
- The resulting image is stored as '521h0435_img_01_12.jpg' using cv.imwrite.
- Create a new image (result_1c_img_02) by setting the pixel values to 0 where the pixel values in the original grayscale image (gray) are equal to 184.

- (result_1c_img_02) is then thresholded using the inverse binary thresholding technique (cv.THRESH_BINARY_INV)
- Pixels with a value of 0 in result_1c_img_02 will be set to 255 in the resulting binary image (thresh_07), and vice versa.the resulting binary image (thresh_07) is saved as '521h0435_img_01_13.jpg' using cv.imwrite.

Method 3:

- Create a new image (result_1c_img_03) by setting the pixel values to 0 where the pixel values in the original grayscale image (gray) are equal to 184.
- Apply a thresholding technique (cv.THRESH_TOZERO) to the image (result_1c_img_03). Pixels with a value below 220 are set to 0, and pixels with a value above 220 retain their original intensity.
- Using binary inverse thresholding (cv.THRESH_BINARY_INV). This creates a binary mask where pixels equal to 0 are set to 255, and pixels above 0 are set to 0.
- The cv.bitwise_not function is used to invert the pixels in thresh_08. This operation essentially replaces the pixels in thresh_08 with their bitwise NOT values where the mask is non-zero.
- The resulting image (result_1c_img_03) is saved as '521h0435_img_01_14.jpg' using cv.imwrite.

1.4 Draw rectangles surrounding each digit in the input image automatically

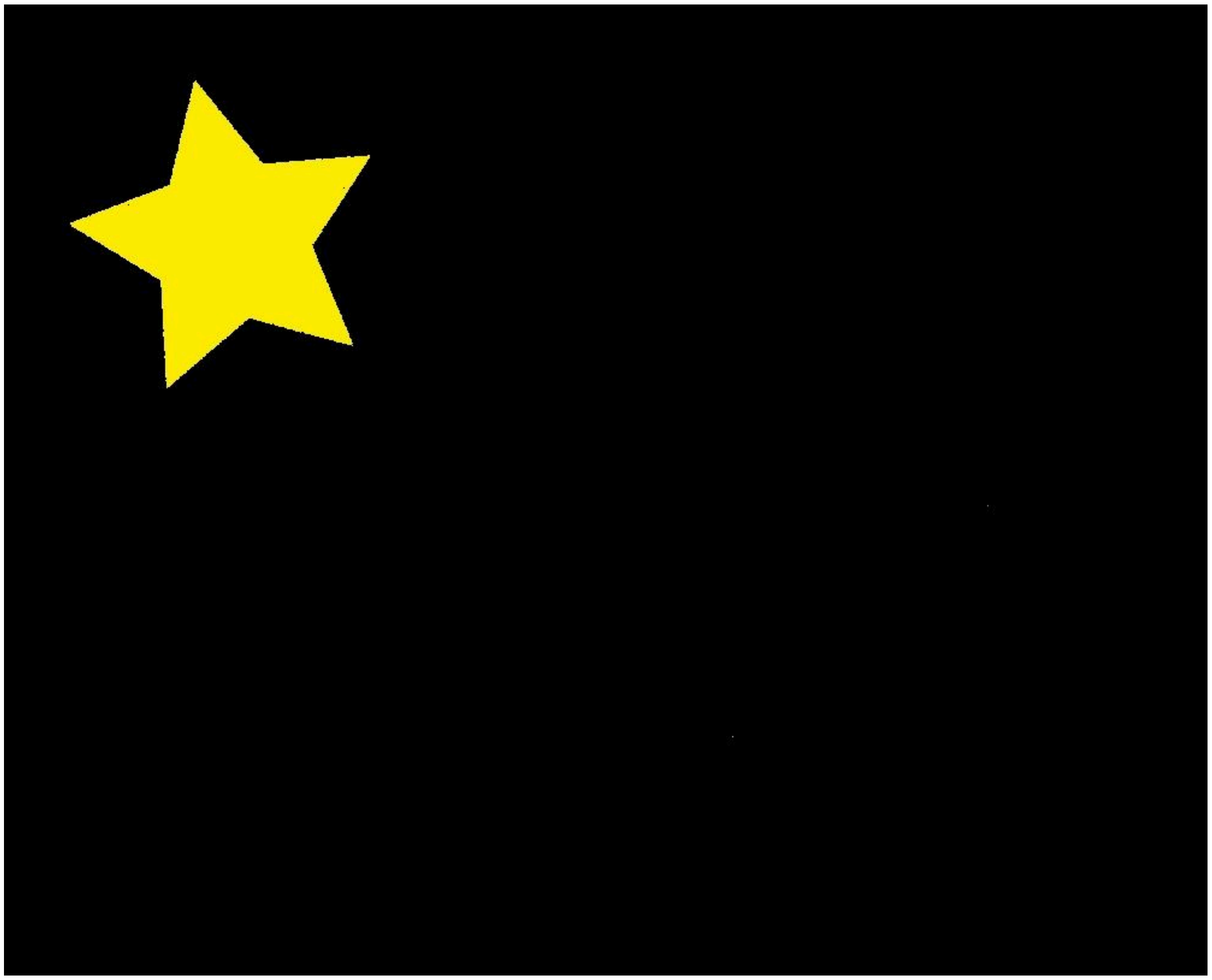
- Converting the input image (img2) to grayscale using cv.cvtColor
- Using cv.threshold with cv.THRESH_BINARY_INV +

cv.THRESH_OTSU to help in creating a binary image where digits are highlighted.

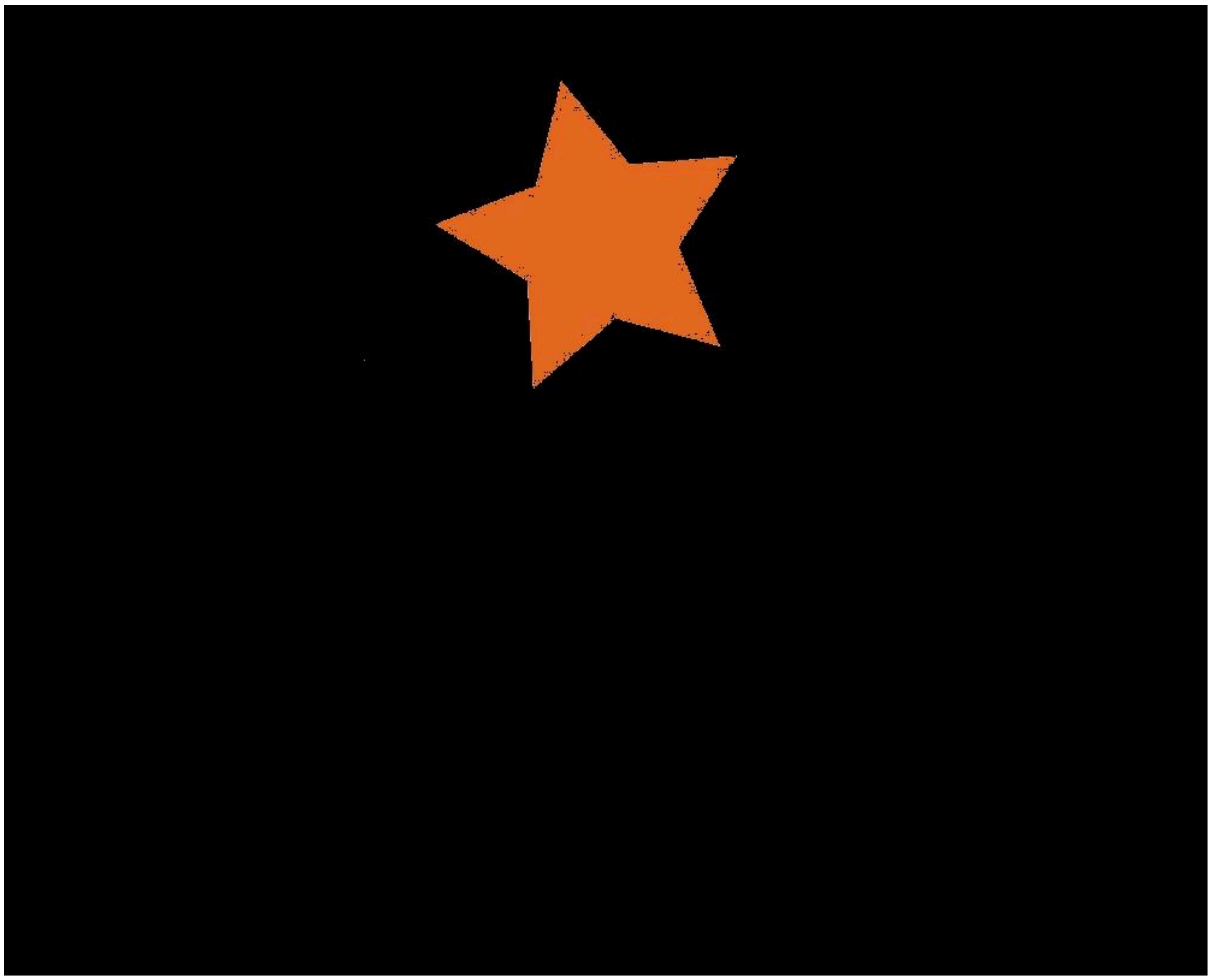
- To eliminate noise and small gaps between digits, a morphological opening operation is performed using a square-shaped kernel (opening_kernel) with cv.morphologyEx.
- Contours are then identified in the processed image using cv.findContours.

- For each identified contour, a bounding rectangle is obtained using cv.boundingRect.
- A green rectangle is drawn around each digit on the original color image (img2) using cv.rectangle.
- The resulting image with rectangles drawn around digits is saved as '521h0435_img_02_01.jpg' using cv.imwrite.

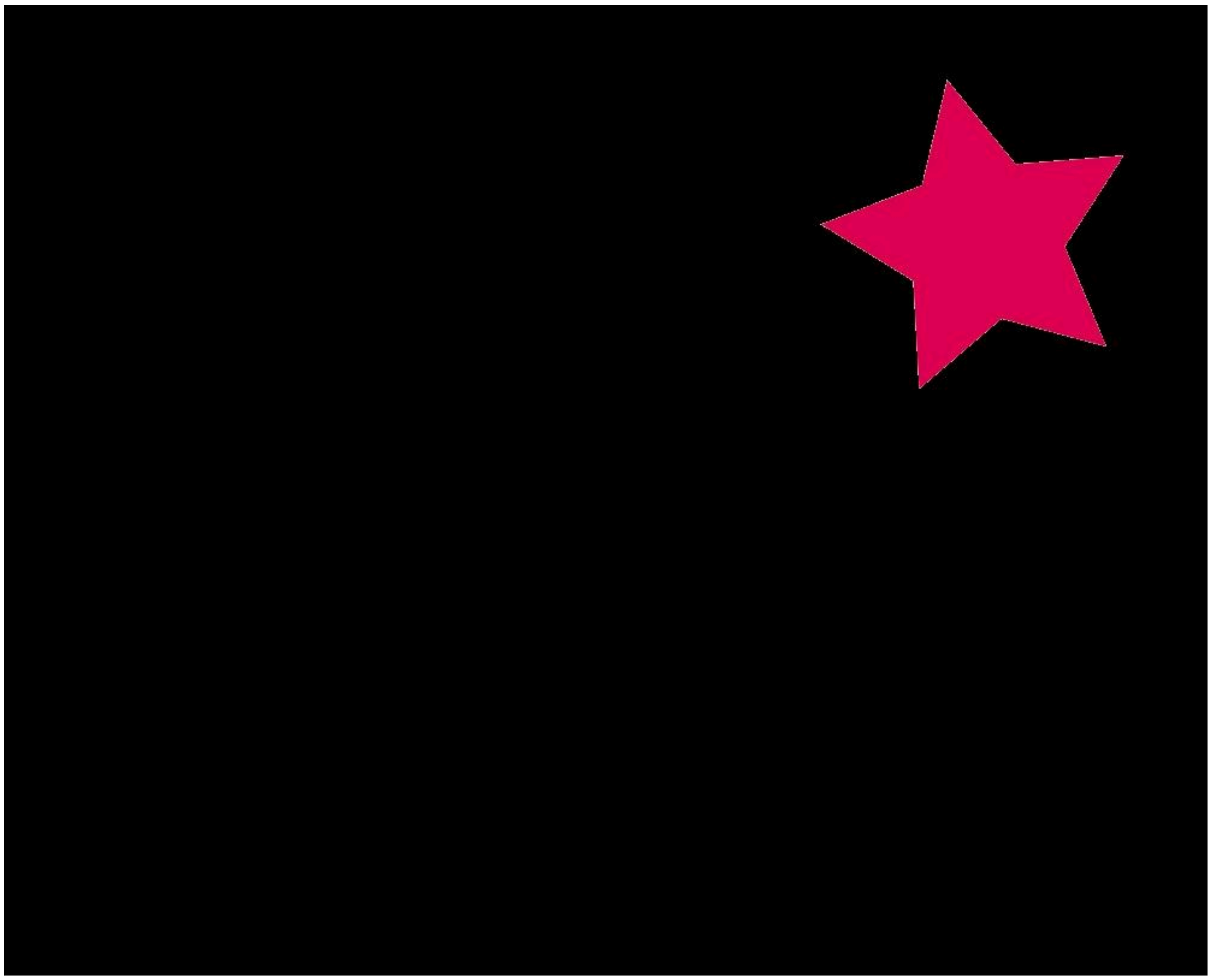
CHAPTER 2: TASK RESULT



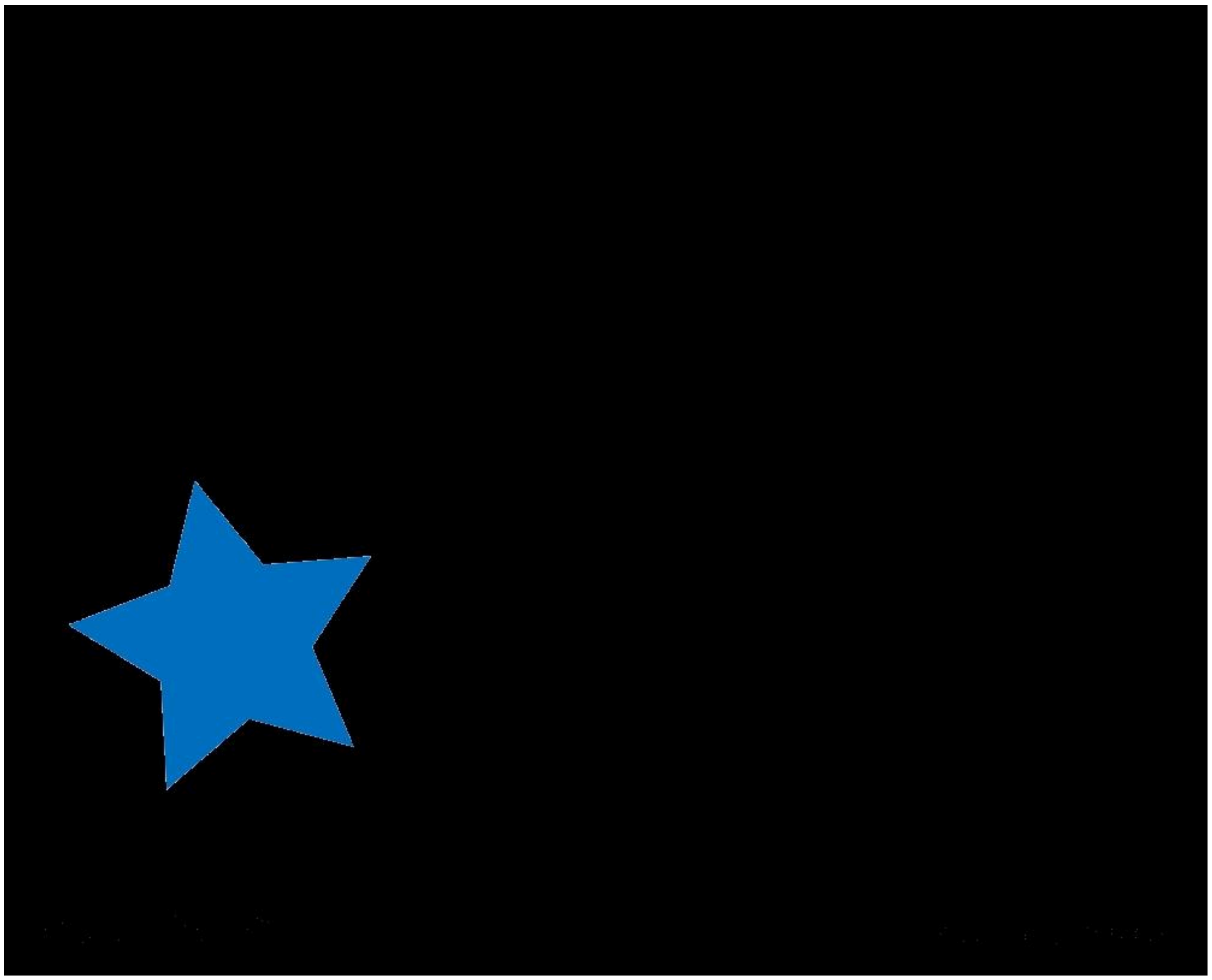
Picture 1. Extract yellow star in the input image.



Picture 2. Extract orange star in the input image.



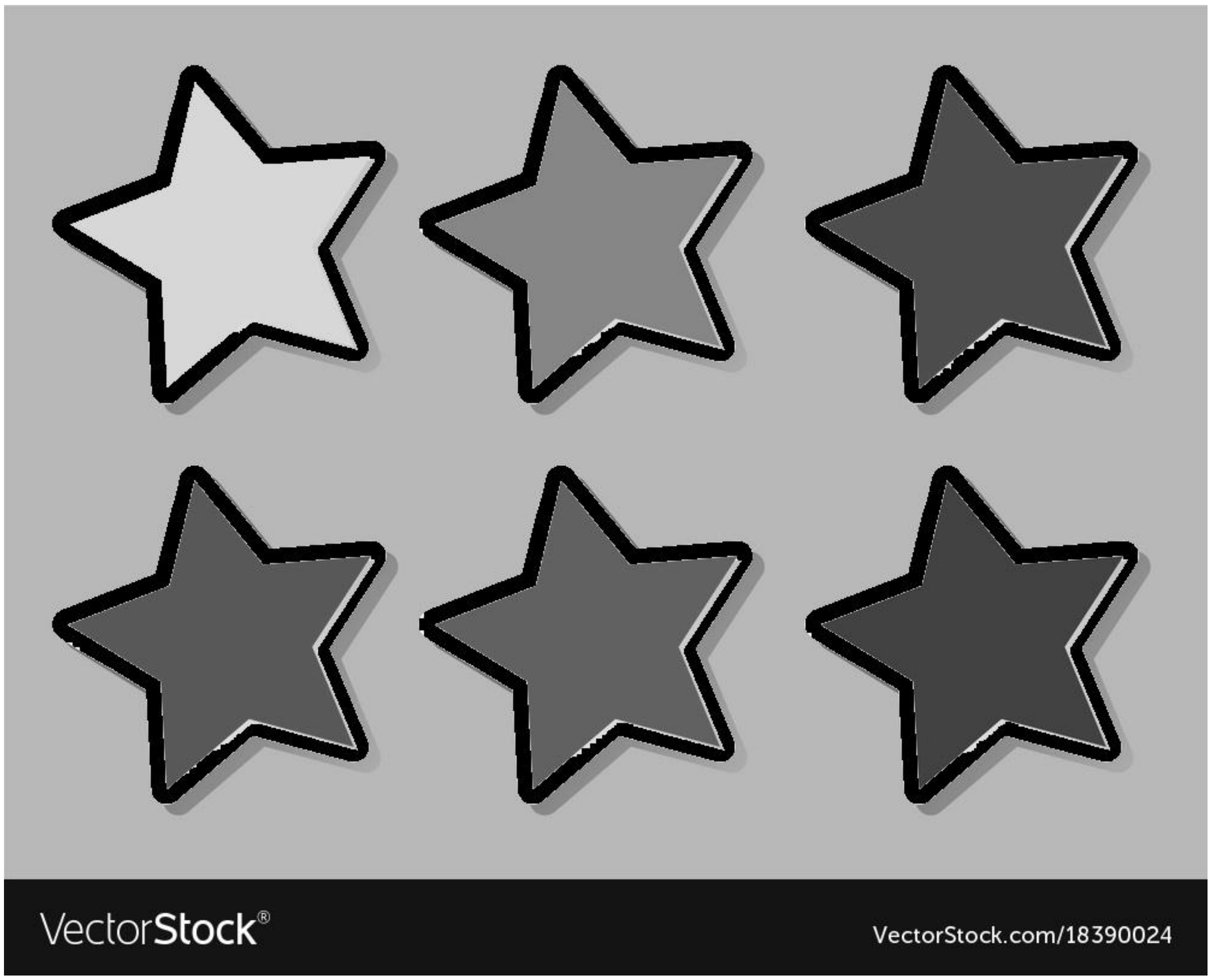
Picture 3. Extract pink star in the input image.



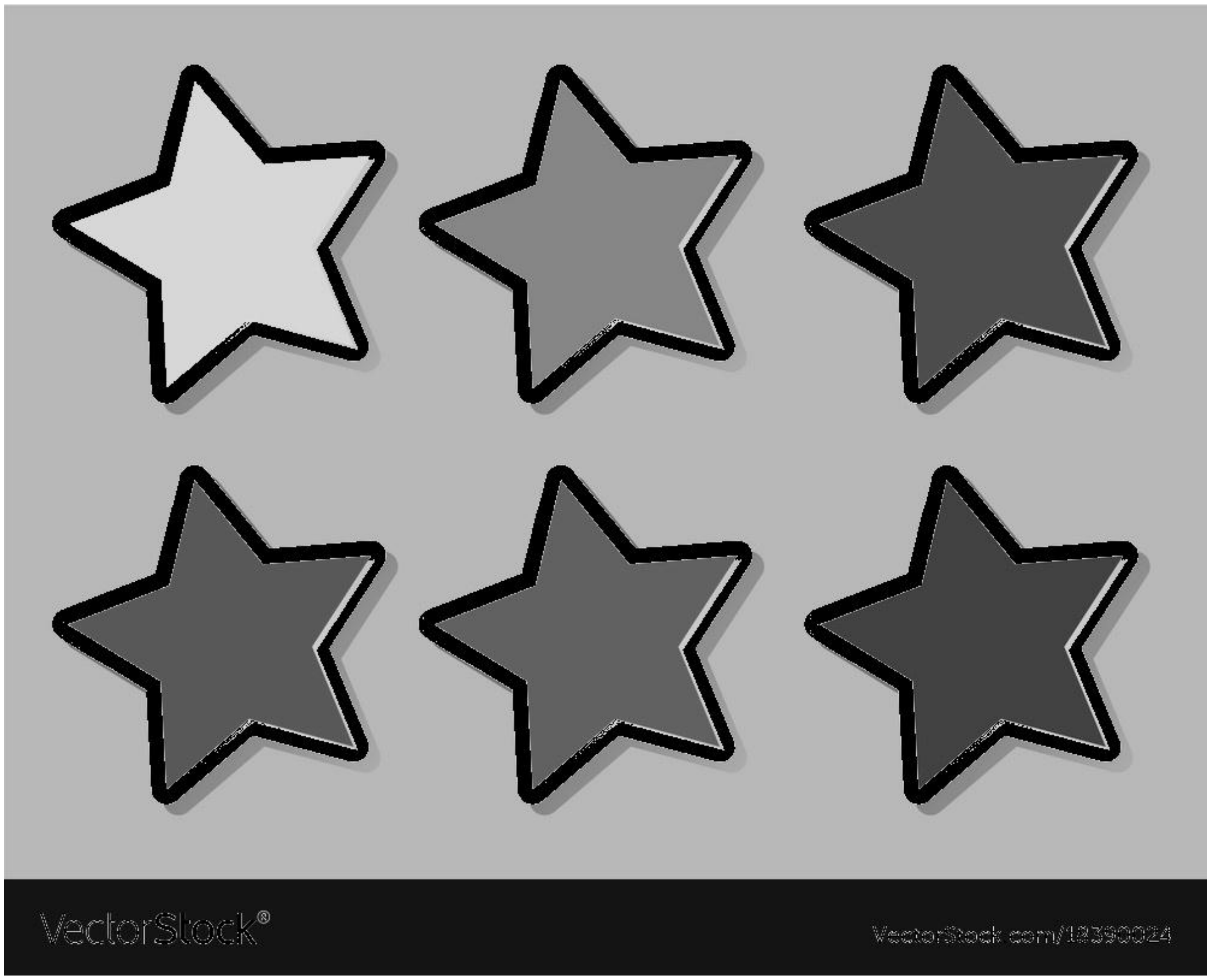
Picture 4. Extract blue star in the input image.

Picture 5. Extract green star in the input image.

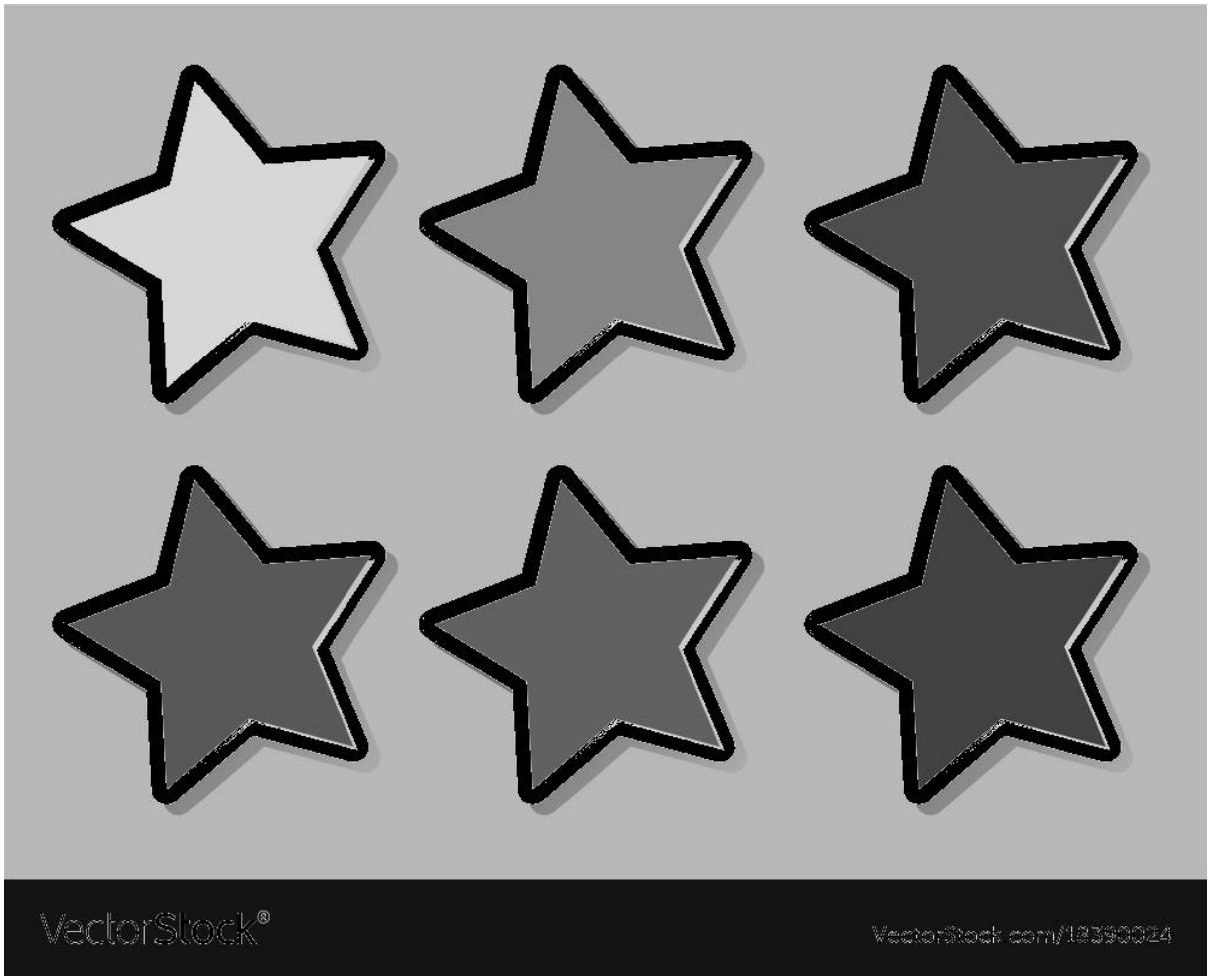
Picture 6. Extract purple star in the input image.



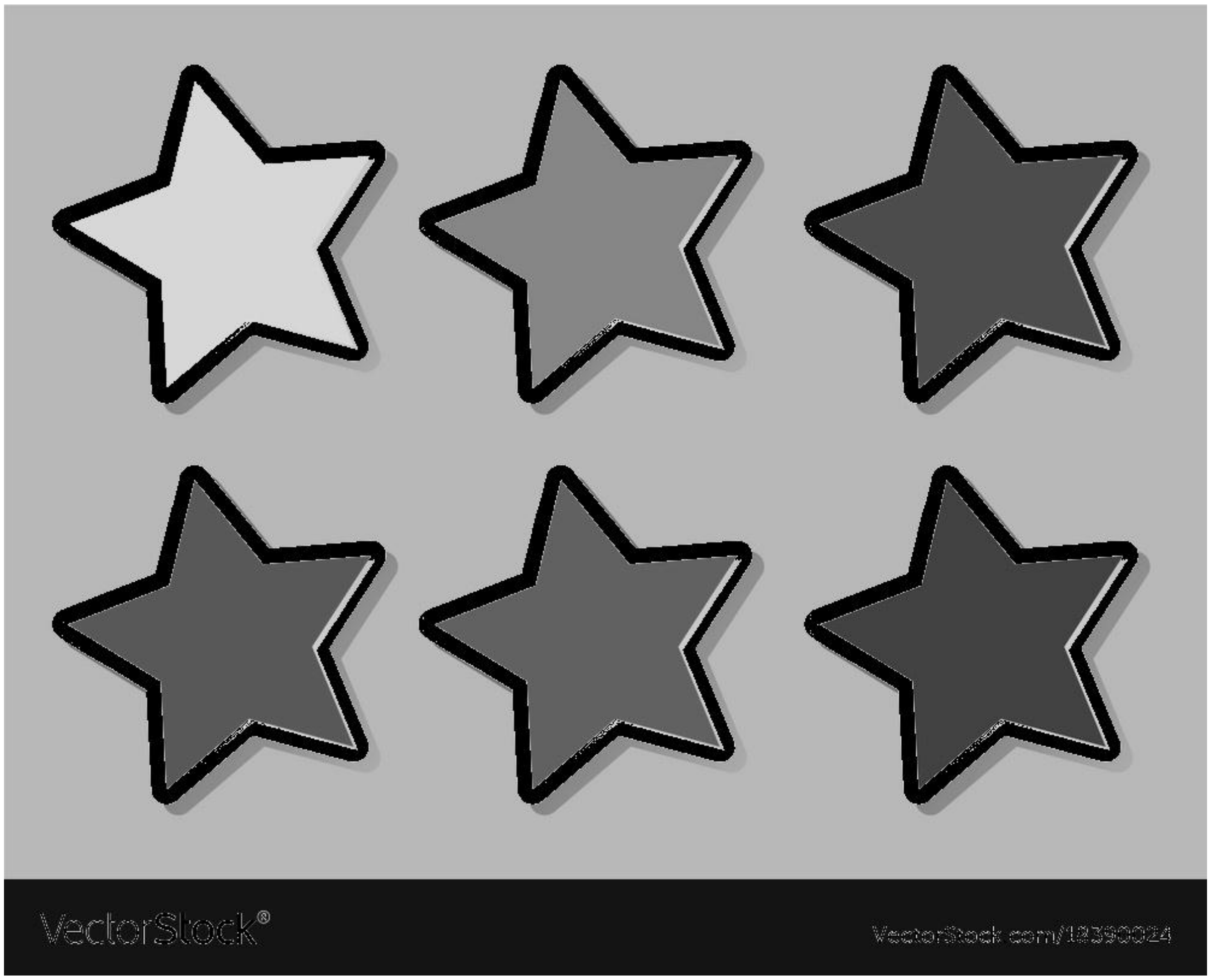
Picture 7. Repaint White borders of all stars to Black color by using cv2.THRESH_BINARY.



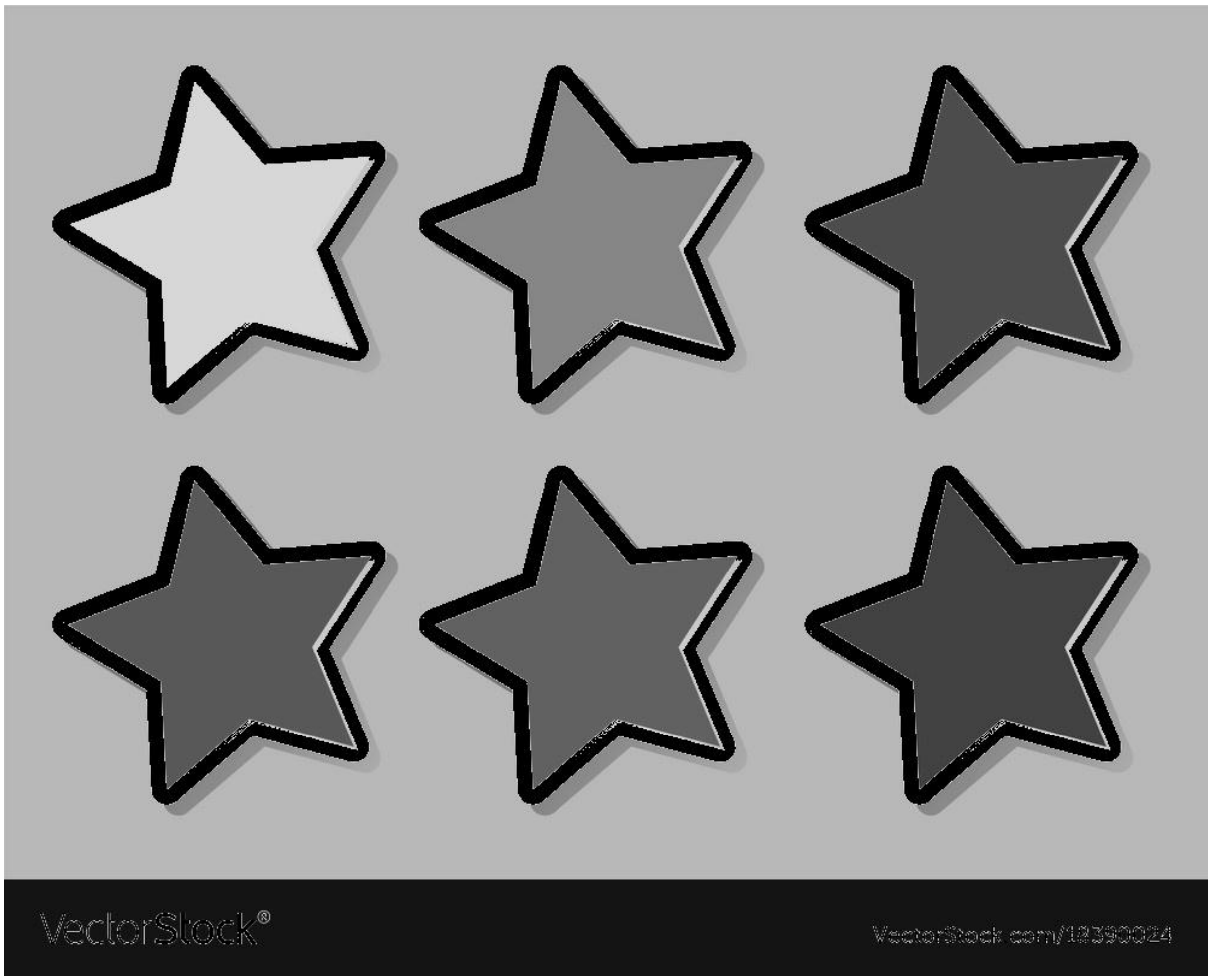
Picture 8. Repaint White borders of all stars to Black color by using cv2.THRESH_TOZERO.



Picture 9. Repaint White borders of all stars to Black color by using cv2.THRESH_BINARY_INV.



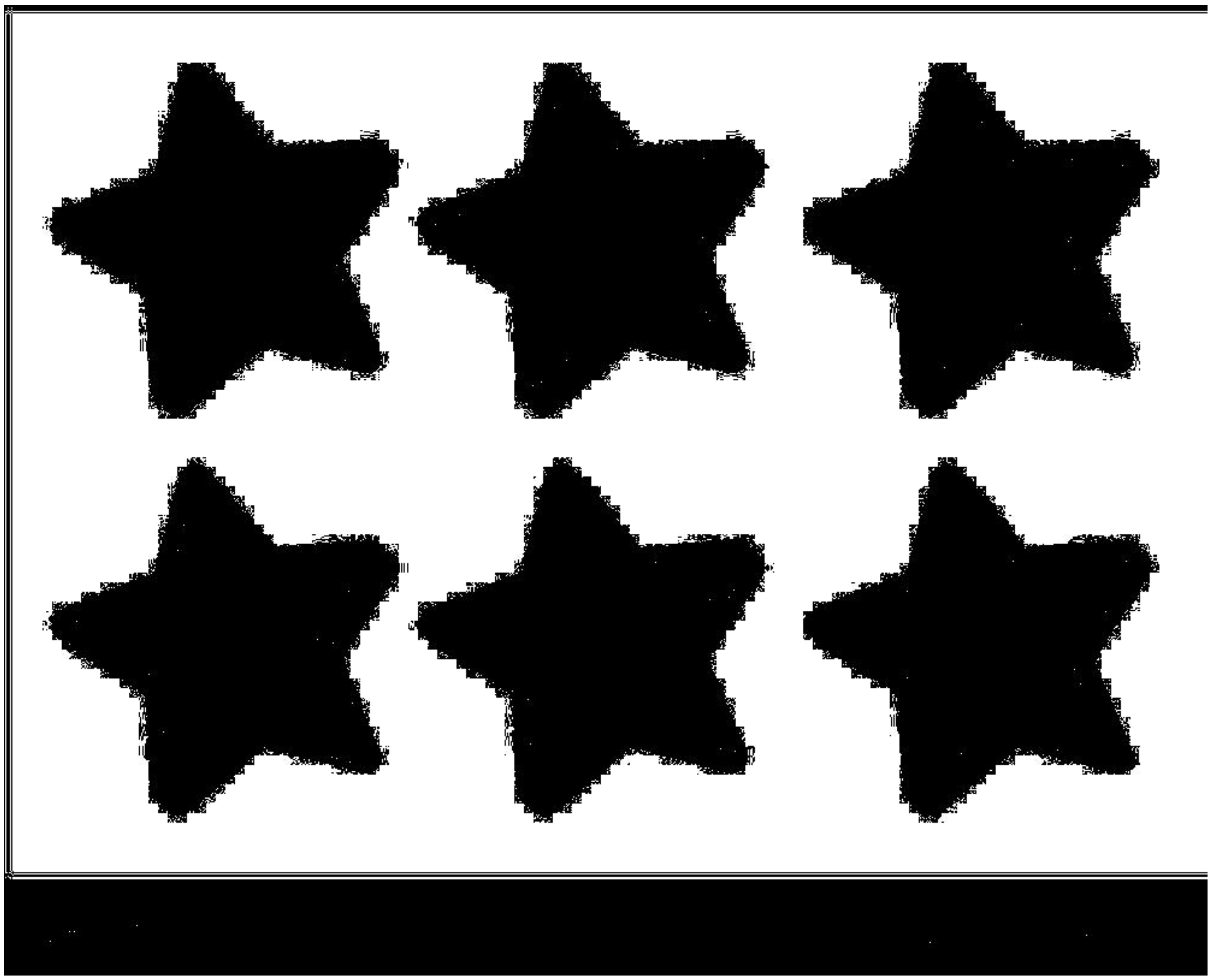
Picture 10. Repaint White borders of all stars to Black color by using cv2.THRESH_TOZERO_INV.



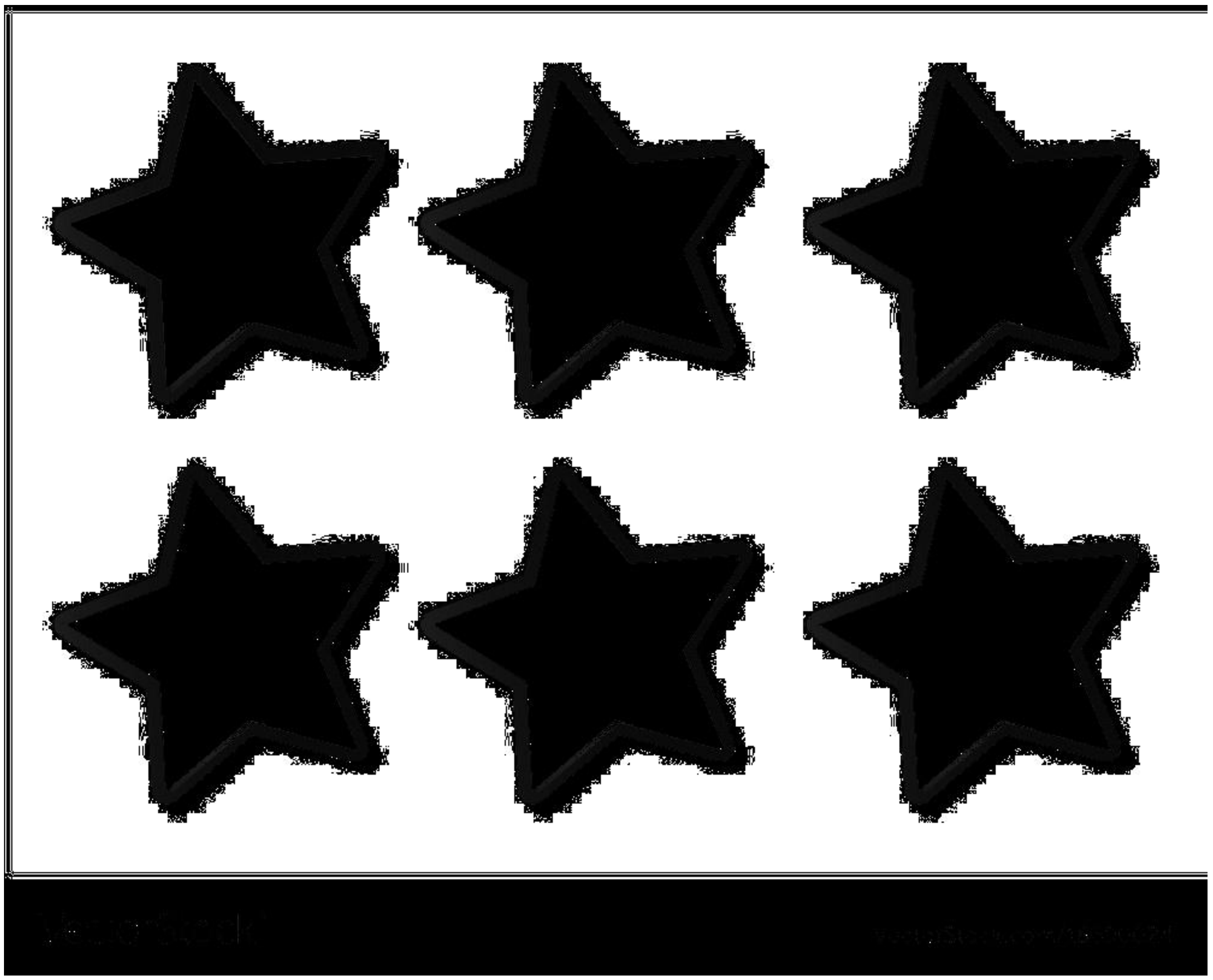
Picture 11. Repaint White borders of all stars to Black color by using cv2.THRESH_TRUNC.

VectorStock
VectorStock.com/18390024

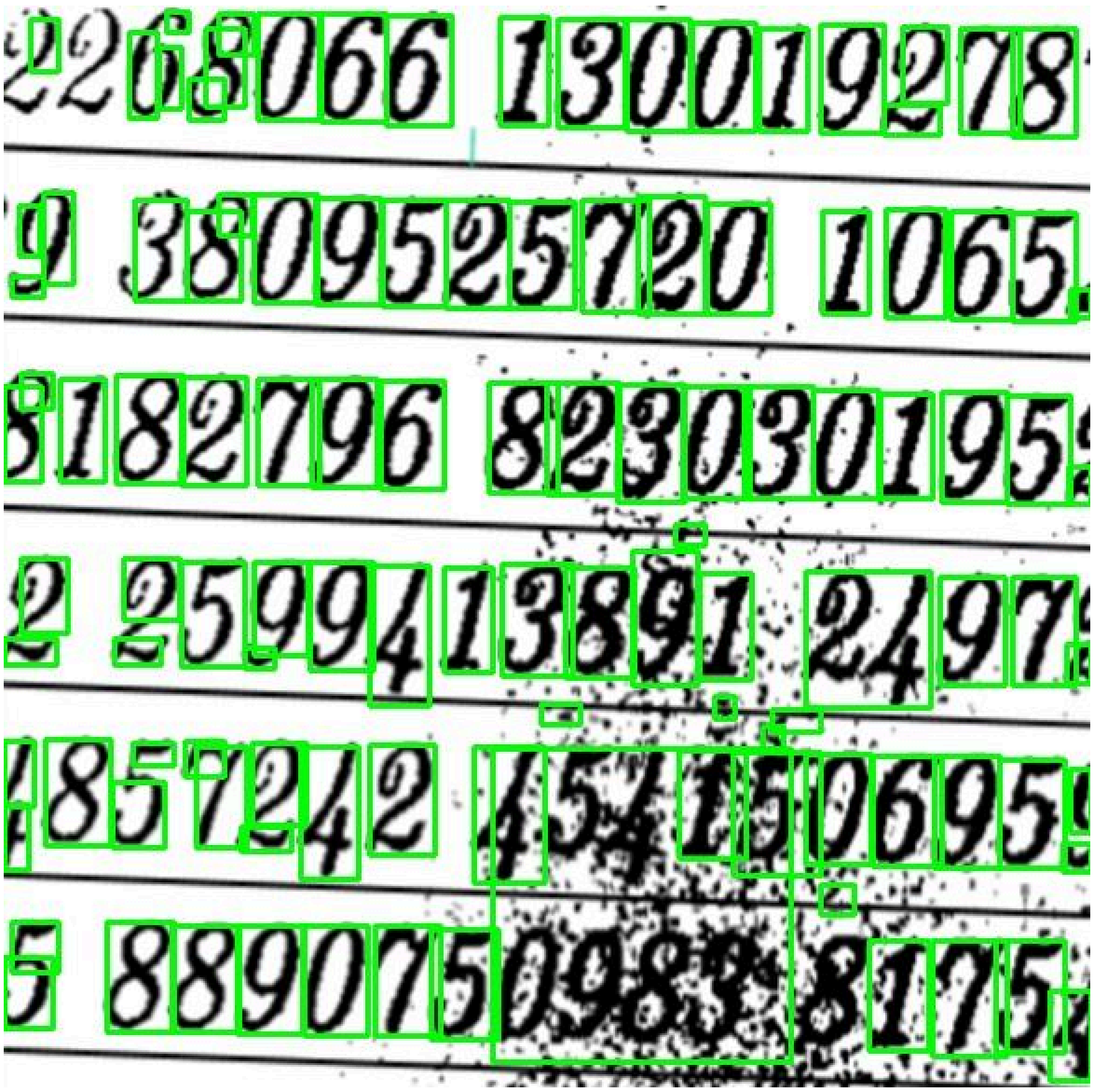
Picture 12. Repaint all stars to Black color by using cv2.THRESH_BINARY, cv2.findContours(), and cv2.drawContours().



Picture 13. Repaint all stars to Black color by using cv2.THRESH_BINARY_INV.



Picture 14. Repaint all stars to Black color by using cv2.THRESH_BINARY_INV, cv2.THRESH_TOZERO, and cv2.bitwise_not().



Picture 15. Draw rectangles surrounding each digit in the input image automatically.