```python
# 24-678 Computer Vision for Engineers
# Ryan Wu (ID:weihuanw)
# PS06-2 Detecting Defective Parts
# Due 11/10/2023 (Fri) 5 pm

# import the necessary packages
import cv2

# defect detection function
def detect_defect(image):
    # convert to grayscale
    gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # convert to binary
    _, dst = cv2.threshold(gray_image, 60, 255, cv2.THRESH_BINARY)

    # dilation
    for contours in range(1):
        dst = cv2.erode(dst, None)

    # erosion
    for contours in range(2):
        dst = cv2.dilate(dst, None)

    # set a threshold for shape matching
    matching_threshold = 1.5

    # set a threshold for filtering out the edge
    max_contour_area = 50000

    # find contours
    cont, _ = cv2.findContours(dst, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

    # contour matching and draw contours
    for contours in range(len(cont)):
        c = cont[contours]
        match_contour = cv2.matchShapes(cont[3], c, cv2.CONTOURS_MATCH_I2, 0)
        if match_contour > matching_threshold and cv2.contourArea(c) < max_contour_area:
            image = cv2.drawContours(image, cont, contours, (0, 0, 255), -1)

    # display the output image
    cv2.imshow('spade-terminal-output image', image)
    cv2.waitKey(0)
    cv2.destroyAllWindows()

    # Save the output image
    cv2.imwrite('spade-terminal-output.png', image)

if __name__ == "__main__":
    input_image = cv2.imread("spade-terminal.png")
    detect_defect(input_image)
```