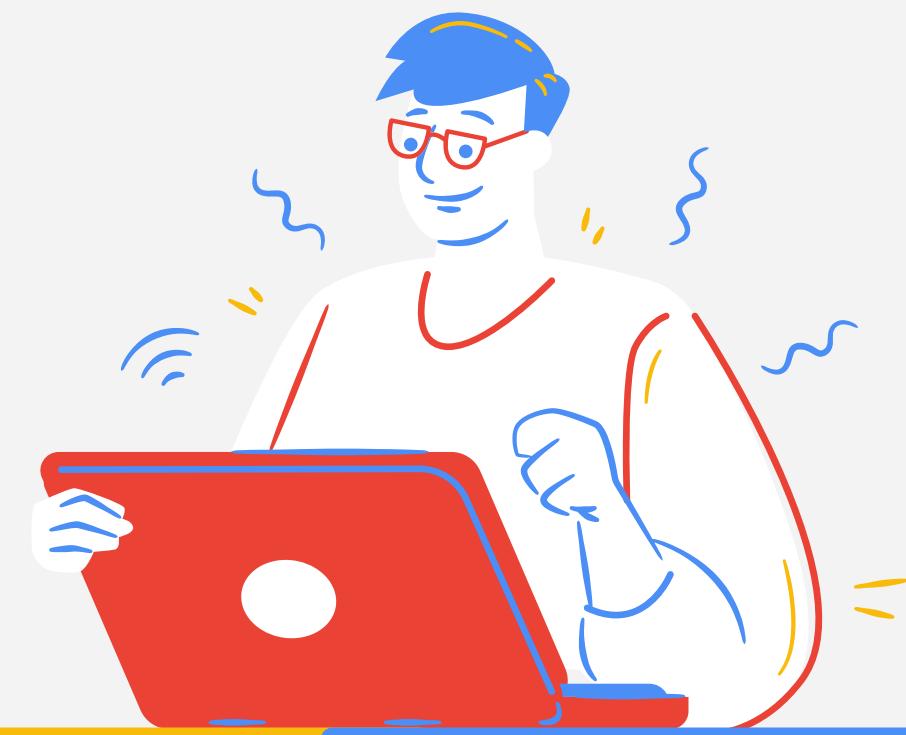


# Deskewing Image Program

DSI206



# Deskewing Image Program



The image shows a side-by-side comparison of a receipt from Whole Foods Market. On the left, the receipt is skewed, appearing tilted. On the right, the same receipt is shown in its original, correctly oriented state. A large red curved arrow points from the skewed receipt to the deskewed one, highlighting the transformation. The receipt includes the store's name, address, and a list of items with their prices.

WHOLE FOODS MARKET - WESTPORT, CT 06880  
399 POST RD WEST - (203) 227-6858

365 BACON LS NP 4.99 F  
365 BROTH CHIC NP 2.19 F  
FLOUR ALMOND NP 18.80 F  
CHKN BRST BNLSS SK NP 18.80 F  
HEAVY CREAM NP 3.39 F  
BALSMC REDUCT NP 6.49 F  
BEEF GRND 85/15 NP 5.04 F  
JUICE COF CASHEW C NP 8.99 F  
DOCS PINT ORGANIC NP 14.49 F  
HNY ALMOND BUTTER NP 9.99 F  
\*\*\*\* TAX .00 BAL 101.33

1200x600

WHOLE FOODS MARKET - WESTPORT, CT 06880  
399 POST RD WEST - (203) 227-6858

365 BACON LS NP 4.99 F  
365 BROTH CHIC NP 2.19 F  
FLOUR ALMOND NP 11.99 F  
CHKN BRST BNLSS SK NP 18.80 F  
HEAVY CREAM NP 3.39 F  
BALSMC REDUCT NP 6.49 F  
BEEF GRND 85/15 NP 5.04 F  
JUICE COF CASHEW C NP 8.99 F  
DOCS PINT ORGANIC NP 14.49 F  
HNY ALMOND BUTTER NP 9.99 F  
\*\*\*\* TAX .00 BAL 101.33

visit



1

2

3

# Deskewing Image Program : Techniques



- Canny Edge Detection
  - Contour
  - Affine Transformation
- 
- + Threshold
  - + OCR

1

2

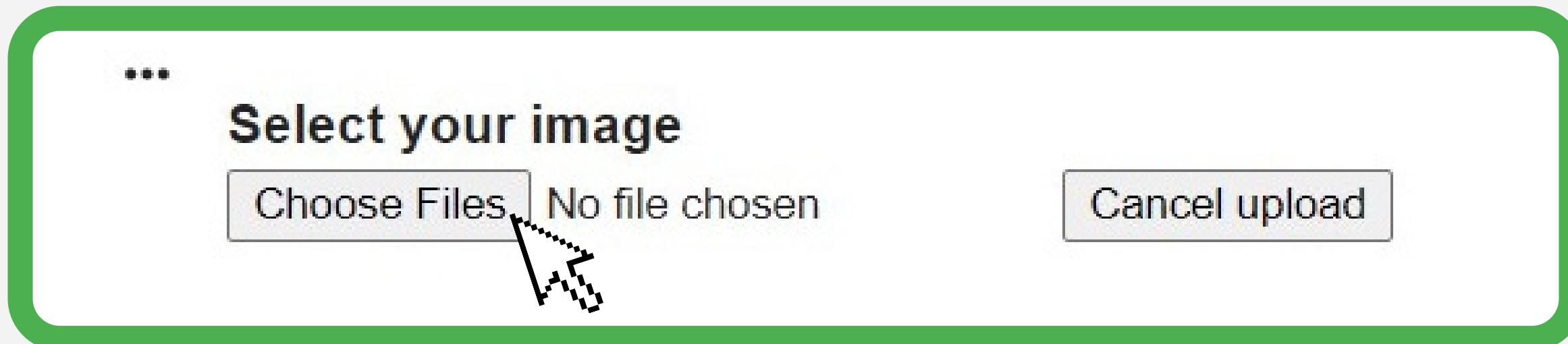
3



# Deskewing Image Program : How to use



- กด Choose File เพื่อเลือกรูปภาพเอกสารที่ต้องการ
- รอแสดงผลลัพธ์



## คำแนะนำ :

- ภาพเอกสารควรเห็นมุมครบกึ่ง 4 มุม ไม่มีอะไรมาบดบังขอบกระดาษ
- ภาพเอกสารควรเห็นตัวอักษรชัดเจนไม่เบลอจนเกินไป
- ควรเป็นภาพอังกฤษ ตัวเลข สัญลักษณ์

1

2

3



## Deskewing Image Program : Prepare Image



4

ขั้นตอนการทำงาน :

จัดเตรียมรูปภาพ โดยการลบ **noise** ออก เพื่อง่ายต่อการนำไปใช้งานต่อ โดยปรับภาพเป็น **grey scale** ปรับให้มี **ความเบลอ** และ **ลด noise ภายในและภายนอกตัวอักษร**

5

```
imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
imgBlur = cv2.GaussianBlur(imgGray, (5,5), 1)
kernel = np.ones((2,2), np.uint8)
opening = cv2.morphologyEx(imgBlur, cv2.MORPH_OPEN, kernel)
closing = cv2.morphologyEx(opening, cv2.MORPH_CLOSE, kernel)
```

6

- code การเตรียมรูปภาพด้วยคำสั่งและเทคนิคต่างๆ



# Deskewing Image Program : **Prepare Image**



ผลลัพธ์ :



4

5

6



# Deskewing Image Program : **Canny Edge Detection**

ขั้นตอนการทำงาน :

เริ่มใช้เทคนิค **Canny Edge Detection**  
โดยกำให้ภาพเป็นใบหน้า และคัดแยกวัตถุออกจากพื้นหลัง

```
edges = cv2.Canny(closing, 20, 240)
imgThreshold = cv2.adaptiveThreshold(edges, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY)
```

- code การทำ Canny Edge

```
imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
imgBlur = cv2.GaussianBlur(imgGray, (5,5), 1)
kernel = np.ones((2,2), np.uint8)
imgOpening = cv2.morphologyEx(imgBlur, cv2.MORPH_OPEN, kernel)
imgClosing = cv2.morphologyEx(imgOpening, cv2.MORPH_CLOSE, kernel)
imgCanny = cv2.Canny(imgClosing, 20, 240)
imgThreshold = cv2.adaptiveThreshold(imgCanny, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY)

cv2_imshow(imgClosing)
cv2_imshow(imgCanny)
cv2_imshow(imgThreshold)
```

- code การรีบูต noise และทำ Canny Edge Detection

4

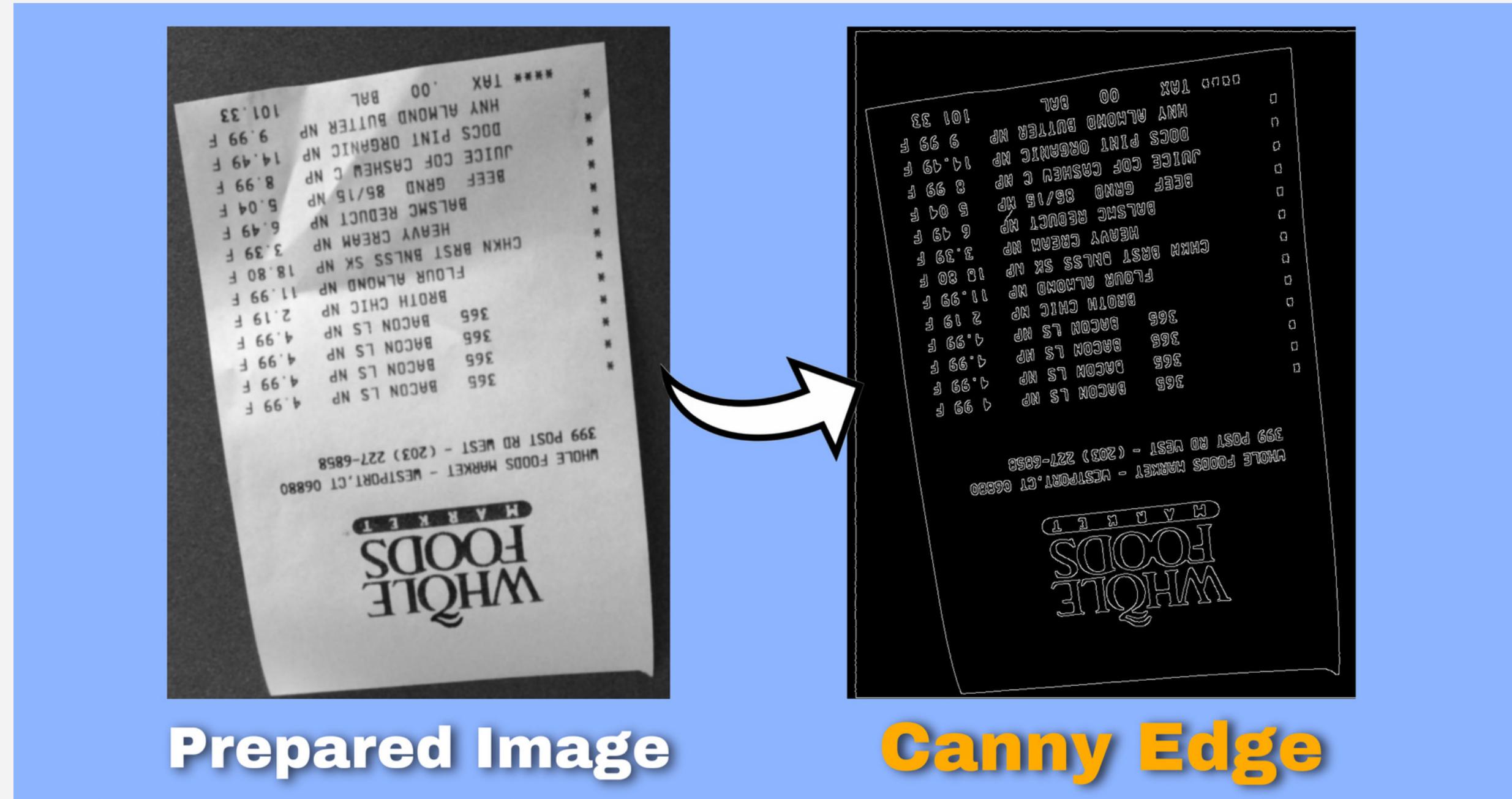
5

6



# Deskewing Image Program : Canny Edge Detection

ผลลัพธ์ :



Output 1

7

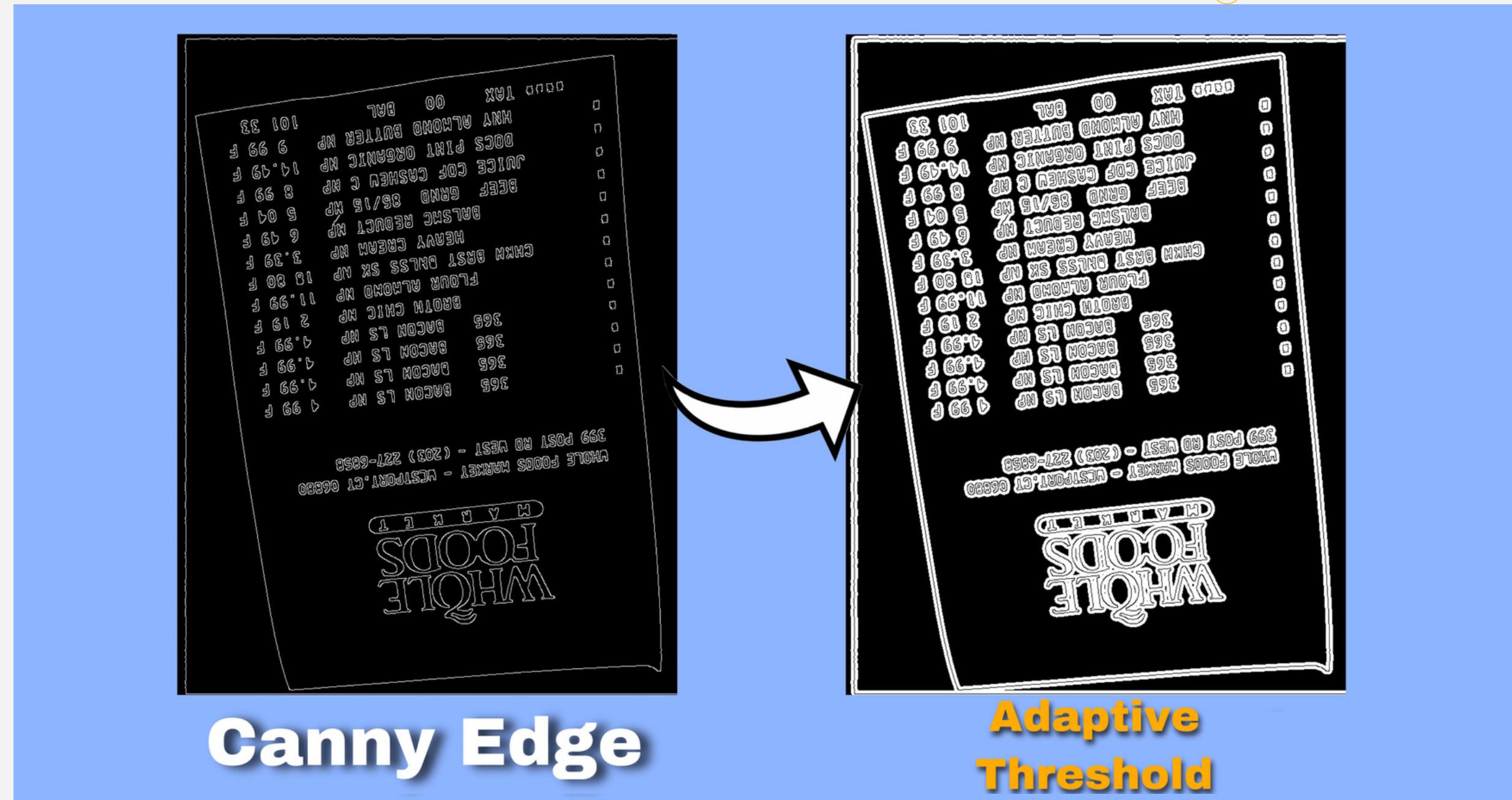
8

9



# Deskewing Image Program : Canny Edge Detection

ผลลัพธ์ :



Output 2

7

8

9



# Deskewing Image Program : **Contour**



7

8

9

ขั้นตอนการทำงาน :

ใช้เทคนิค **Contour** เพื่อแสดงรูปร่างหรือจุดต่อเนื่องกึ่งหมด ที่มีความเข้มเก่ากัน โดยการหาค่อนกัวร์ของภาพและ sorted เลือกใช้ค่อนกัวร์ที่ใหญ่ที่สุดซึ่งอนุมานว่าเป็น ขอบกระดาษ แล้วนำมาเช็คว่ามีรูปร่างเป็นสี่เหลี่ยมหรือไม่ หากเป็นสี่เหลี่ยมก็จะบันทึกค่า จุดมุนกึ่ง 4 มุมไว้ และวัดเส้นเชื่อมต่อเพื่อแสดงขอบกระดาษ





# Deskewing Image Program : Contour

## ขั้นตอนการทำงาน :

```
imgContours = img.copy()
contours, hierarchy = cv2.findContours(imgThreshold, cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)
contours = sorted(contours, key = cv2.contourArea, reverse = True)[:5]
```

- code สำหรับหาค่อนกัวร์ และ Sorted ค่อนกัวร์

```
for i in contours:
    peri = cv2.arcLength(i, True)
    approx = cv2.approxPolyDP(i, 0.02 * peri, True)
    if len(approx) == 4:
        cnts = approx
        break
cv2.drawContours(imgContours, [cnts], -1, (0,255,0), 3)
```

- code สำหรับเช็ครูปร่างค่อนกัวร์ เพื่อเก็บค่าจุด มุมและวัดเส้นเชื่อมมุมทั้ง 4

```
#Contours
imgContours = img.copy()
contours, hierarchy = cv2.findContours(imgThreshold, cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)
contours = sorted(contours, key = cv2.contourArea, reverse = True)[:5]
for i in contours:
    peri = cv2.arcLength(i, True)
    approx = cv2.approxPolyDP(i, 0.02 * peri, True)
    if len(approx) == 4:
        cnts = approx
        break
cv2.drawContours(imgContours, [cnts], -1, (0,255,0), 3)
cv2_imshow(imgContours)
```

- code สำหรับเทคนิค Contour

10

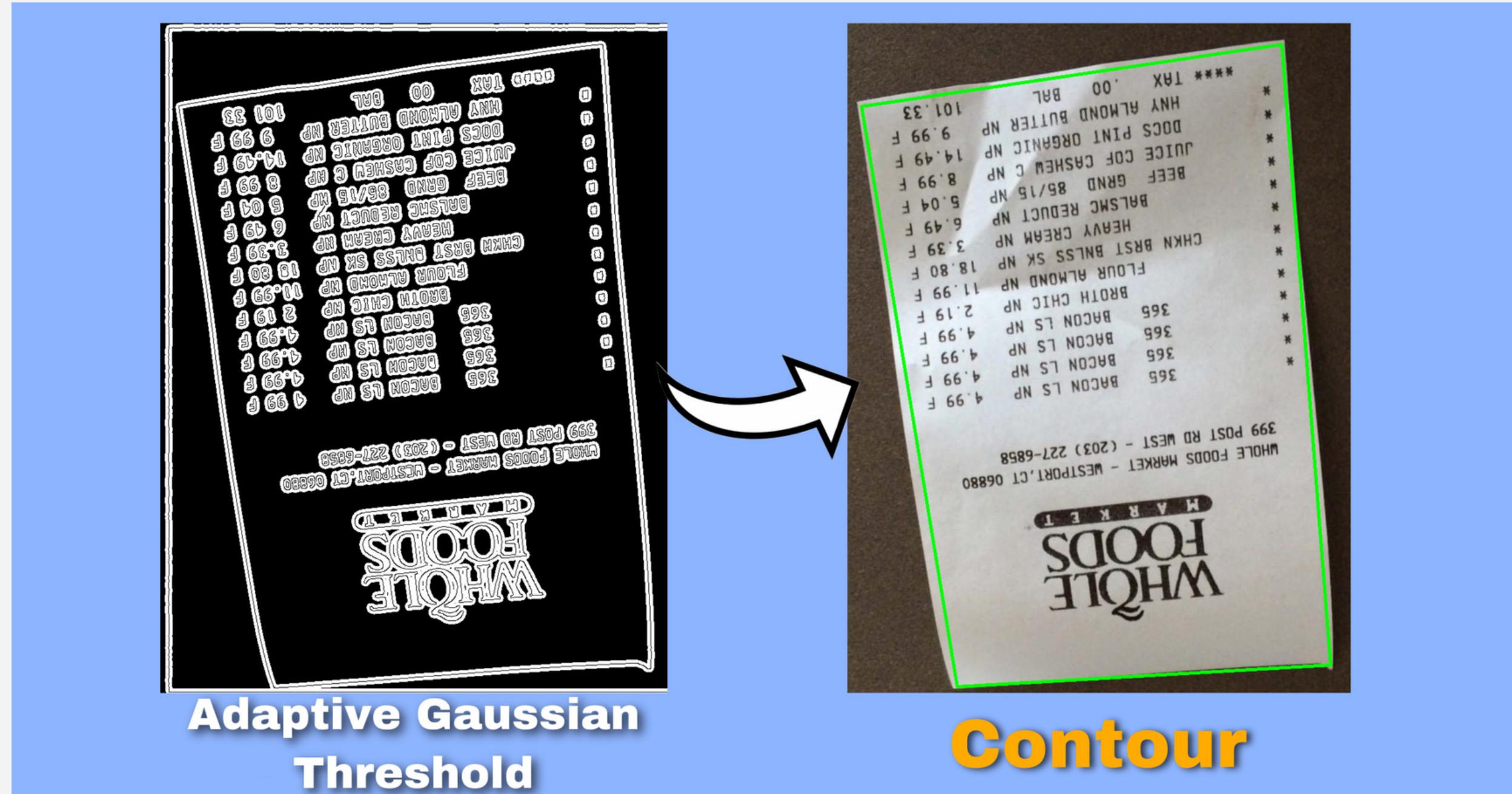
11

12



# Deskewing Image Program : **Contour**

ผลลัพธ์ :



10

11

12



# Deskewing Image Program : **Affine Transformation**



ขั้นตอนการทำงาน :

**เปลี่ยนหน้ากระดาษให้เป็นหน้าตรง โดยใช้เทคนิค Affine Transformation** จากนั้นจะทำการคำนวณ transformation matrix( $M$ ) โดยเปลี่ยนจากจุดมุ่ง contour 4 มุม เป็นจุดมุ่งของรูปภาพใหม่ที่จัดตรงแล้ว

10

11

12





# Deskewing Image Program : Affine Transformation



13

ขั้นตอนการทำงาน :

```
def reorder(pts):
    pts = pts.reshape((4, 2))
    rect = np.zeros((4, 2), dtype=np.float32)
    add = pts.sum(axis = 1)
    rect[0] = pts[np.argmin(add)]
    rect[2] = pts[np.argmax(add)]

    diff = np.diff(pts, axis=1)
    rect[1] =pts[np.argmin(diff)]
    rect[3] = pts[np.argmax(diff)]
    return rect

rect = reorder(cnts)
(tl, tr, br, bl) = rect

widthT = np.sqrt(((tr[0] - tl[0]) ** 2) + ((tr[1] - tl[1]) ** 2))
widthB = np.sqrt(((br[0] - bl[0]) ** 2) + ((br[1] - bl[1]) ** 2))

heightL = np.sqrt(((tl[0] - bl[0]) ** 2) + ((tl[1] - bl[1]) ** 2))
heightR = np.sqrt(((tr[0] - br[0]) ** 2) + ((tr[1] - br[1]) ** 2))

maxWidth = max(int(widthB), int(widthT))
maxHeight = max(int(heightR), int(heightL))

rect = np.float32(rect)
dst = np.array([[0, 0], [maxWidth - 1, 0], [maxWidth - 1, maxHeight - 1], [0, maxHeight - 1]], dtype = "float32")
```

14

15

- code กํา Transformation Matrix(M)



# Deskewing Image Program : Affine Transformation



ขั้นตอนการดำเนิน :

```
matrix = cv2.getPerspectiveTransform(rect, dst)
imgWarp = cv2.warpPerspective(img, matrix, (maxWidth, maxHeight))
cv2_imshow(imgWarp)
```

- code การแปลงกระดาษให้หน้าตรง

13

```
def reorder(pts):
    pts = pts.reshape((4, 2))
    rect = np.zeros((4, 2), dtype=np.float32)
    add = pts.sum(axis = 1)
    rect[0] = pts[np.argmin(add)]
    rect[2] = pts[np.argmax(add)]

    diff = np.diff(pts, axis=1)
    rect[1] =pts[np.argmin(diff)]
    rect[3] = pts[np.argmax(diff)]
    return rect

rect = reorder(cnts)
(tl, tr, br, bl) = rect

widthT = np.sqrt(((tr[0] - tl[0]) ** 2) + ((tr[1] - tl[1]) ** 2))
widthB = np.sqrt(((br[0] - bl[0]) ** 2) + ((br[1] - bl[1]) ** 2))

heightL = np.sqrt(((tl[0] - bl[0]) ** 2) + ((tl[1] - bl[1]) ** 2))
heightR = np.sqrt(((tr[0] - br[0]) ** 2) + ((tr[1] - br[1]) ** 2))

maxWidth = max(int(widthB), int(widthT))
maxHeight = max(int(heightR), int(heightL))

rect = np.float32(rect)
dst = np.array([[0, 0], [maxWidth - 1, 0], [maxWidth - 1, maxHeight - 1], [0, maxHeight - 1]], dtype = "float32")

matrix = cv2.getPerspectiveTransform(rect, dst)
imgWarp = cv2.warpPerspective(img, matrix, (maxWidth, maxHeight))
cv2_imshow(imgWarp)
```

- code การทำ Affine Transformation

14

15



# Deskewing Image Program : Affine Transformation



ผลลัพธ์ :



13

14

15



## Deskewing Image Program : Threshold



16

ขั้นตอนการทำงาน :

แปลงภาพให้มีความชัดเจนมากขึ้น โดยมีลักษณะพื้นหลังเป็นสีขาว  
และตัวอักษรเป็นสีดำ ชัดขึ้นมา ลักษณะคล้ายการถ่ายเอกสาร โดยใช้คำสั่ง  
**Local Threshold**

17

```
imgWarp = cv2.cvtColor (imgWarp, cv2.COLOR_BGR2GRAY)
T = threshold_local(imgWarp, 11, offset = 10, method = "gaussian")
imgWarp = (imgWarp > T).astype("uint8") * 255
cv2_imshow(imgWarp)
```

18

- code การแปลงภาพให้ลักษณะคล้ายกับการถ่ายเอกสาร



# Deskewing Image Program : Threshold

ผลลัพธ์ :



16

17

18



# Deskewing Image Program : OCR



16

ขั้นตอนการทำงาน :

ใช้**เทคนิค OCR** ในการ**หมุนรูปหน้ากระดาษให้ถูกต้อง** โดยจะหาองศาที่กระดาษหมุนผิดไป และแก้ไขให้หมุนกลับมาในองศาที่ถูกต้อง **โดยใช้ OSD ใน tesseract** ตรวจจับตัวอักษรว่าอยู่ในองศาที่ถูกต้องหรือยัง

17

```
import math
from typing import Tuple, Union
def rotate(image: np.ndarray, angle: float, background: Union[int, Tuple[int, int, int]]) -> np.ndarray:
    old_width, old_height = image.shape[:2]
    angle_radian = math.radians(angle)
    width = abs(np.sin(angle_radian) * old_height) + abs(np.cos(angle_radian) * old_width)
    height = abs(np.sin(angle_radian) * old_width) + abs(np.cos(angle_radian) * old_height)

    image_center = tuple(np.array(image.shape[1::-1]) / 2)
    rot_mat = cv2.getRotationMatrix2D(image_center, angle, 1.0)
    rot_mat[1, 2] += (width - old_width) / 2
    rot_mat[0, 2] += (height - old_height) / 2
    return cv2.warpAffine(image, rot_mat, (int(round(height)), int(round(width))), borderValue=background)
```

18

- code สำหรับหมุนหน้ากระดาษให้กลับมาถูกต้อง



# Deskewing Image Program : OCR

ขั้นตอนการทำงาน :

19

```
while True:  
    osd_rotated_image = pytesseract.image_to_osd(imgWarp)  
    print( osd_rotated_image)  
    # using regex we search for the angle(in string format) of the text  
    angle_rotated_image = re.search('(?=<Rotate: )\d+', osd_rotated_image).group(0)  
  
    if (angle_rotated_image == '0'):  
        imgWarp = imgWarp  
        # break the loop once we get the correctly deskewed image  
        break  
    elif (angle_rotated_image == '90'):  
        imgWarp = rotate(imgWarp,90,(255,255,255)) # rotate(image,angle,background_color)  
        continue  
    elif (angle_rotated_image == '180'):  
        imgWarp = rotate(imgWarp,180,(255,255,255))  
        continue  
    elif (angle_rotated_image == '270'):  
        imgWarp = rotate(imgWarp,90,(255,255,255))  
        continue  
    cv2_imshow(imgWarp)
```

20

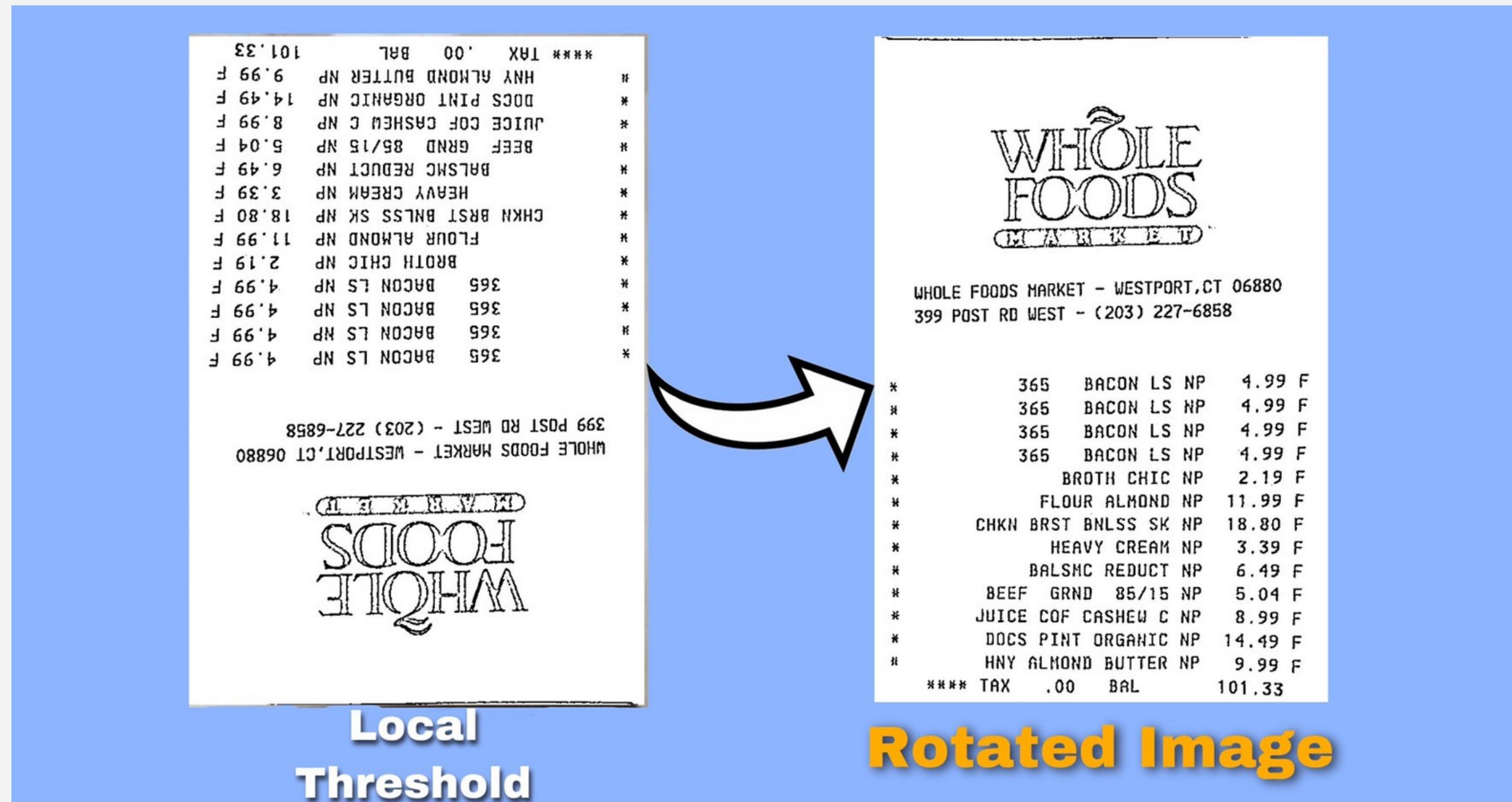
21

- code สำหรับหาองศาที่ภาพหมุนผิดไป



# Deskewing Image Program : OCR

ผลลัพธ์ :



19

20

21



# Deskewing Image Program : OCR



ขั้นตอนการทำ :

การทำ **OCR** โดยใช้ **pytesseract** ตรวจจับข้อความในรูปเอกสาร  
และแปลงเป็นข้อความ และแสดง **text** นี้ออกมาได้

```
imgWarp = cv2.cvtColor(imgWarp, cv2.COLOR_BGR2RGB)
text = pytesseract.image_to_string(imgWarp)
print(text)
```

- code แปลงข้อความในรูปภาพ เป็นข้อความ

19

20

21



22

# Deskewing Image Program : OCR

ผลลัพธ์ :

The image shows a 'Rotated Image' of a Whole Foods Market receipt on the left and its corresponding 'Text' output on the right. A large black curved arrow points from the receipt to the text.

**Rotated Image:**

WHOLE FOODS MARKET - WESTPORT, CT 06880  
399 POST RD WEST - (203) 227-6858

	ITEM	SIZE	PRICE	TYPE
*	365	BACON LS	NP 4.99	F
*	365	BACON LS	NP 4.99	F
*	365	BACON LS	NP 4.99	F
*	365	BACON LS	NP 4.99	F
*		BROTH CHIC	NP 2.19	F
*		FLOUR ALMOND	NP 11.99	F
*	CHKN BRST	BNLSS SK	NP 18.80	F
*		HEAVY CREAM	NP 3.39	F
*		BALSMC REDUCT	NP 6.49	F
*		BEEF GRND 85/18	NP 5.04	F
*	JUICE COF CASHEW C	NP 8.99	F	
*	DOCS PINT ORGANIC	NP 14.49	F	
*	HNY ALMOND BUTTER	NP 9.99	F	
**** TAX .00	BAL	101.33		

**Text:**

WHOLE FOODS  
Gaye ee Dp  
WHOLE FOODS MARKET - WESTPORT.CT 06880  
399 POST RD WEST - (203) 227-6858  
365 BACON LS NP 4.99  
BROTH CHIC NP 2.19  
FLOUR ALMOND NP 11.99  
CHKN BRST BNLSS SK NP 18.80  
HEAVY CREAM NP 3.39  
BALSMC REDUCT NP 6.49  
BEEF GRND 85/18 NP 5.04  
JUICE COF CASHEW € NP 8.99  
DOCS PINT ORGANIC NP 14.49  
HNY ALMOND BUTTER NP 9.99  
eHRH TAX .00 BAL 101,33  
rAWDIADANTAN AAAs

23

24



# Deskewing Image Program : Team



นางสาวณัฐนรี บัวผัด  
6309658232



นางสาวรัตน์ชนก ประคำมินทร์  
6309658778



นายกฤตภาส บุญกระพือ  
6309658869



นายวัฒนชาติ เกื้อพะระ  
6309659164