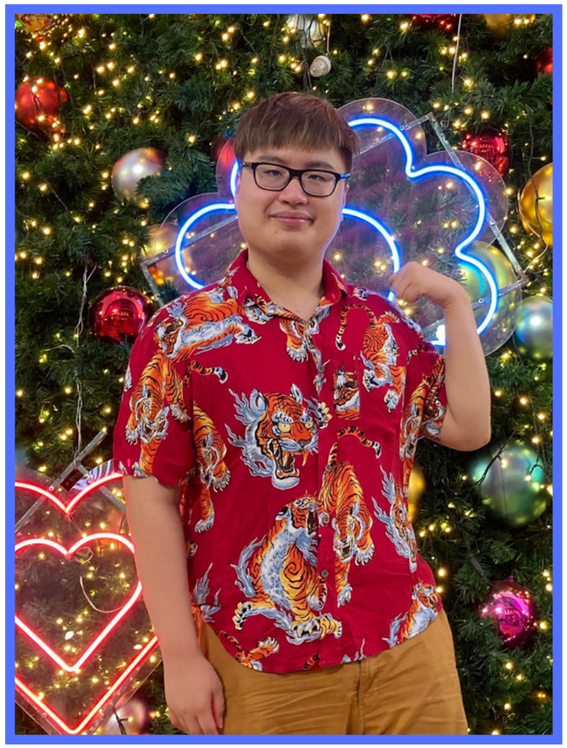
PROJECT
DIGITAL IMAGE

MEMBERS

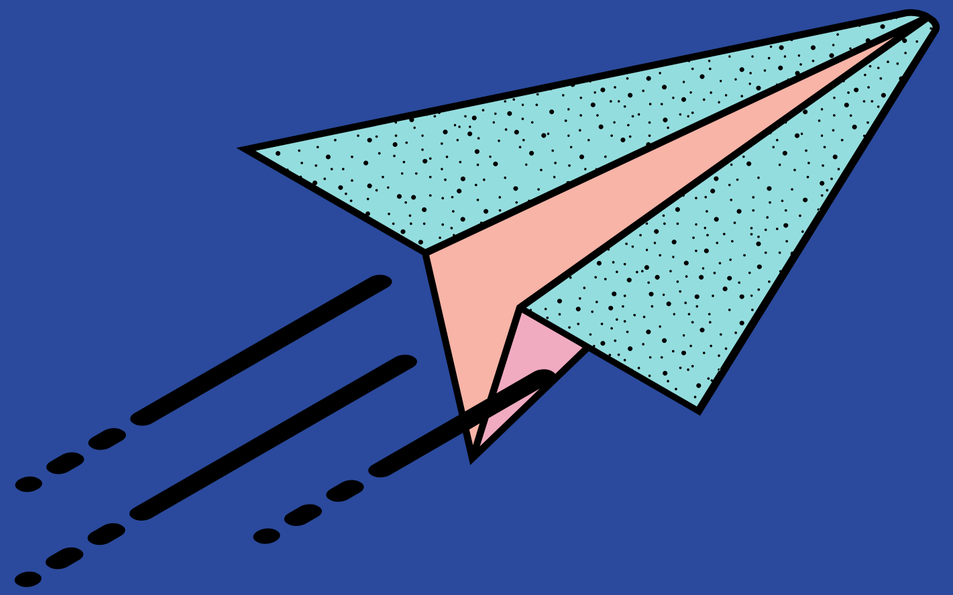


ศุภวิชญ์ ถาวรพิชญไชย
MEMBER 6231364021

ธนิน เศวตกิติธรรม
MEMBER 6230238421

1

INTRODUCTION



PROJECT TOPIC

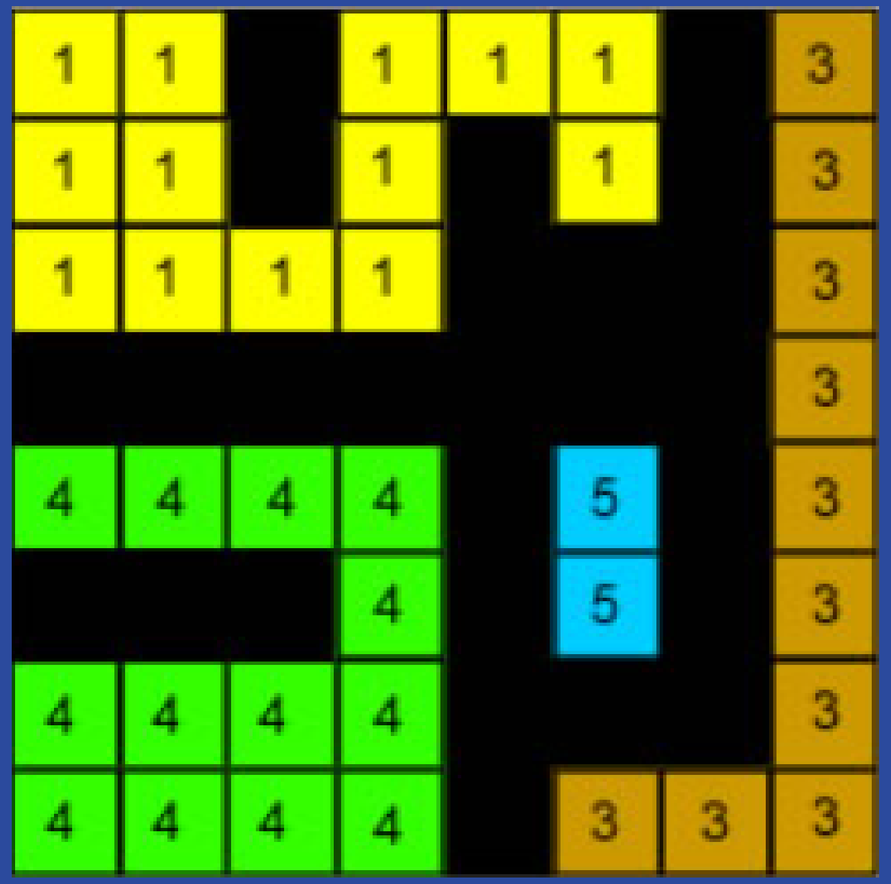
COUNTING ABC BISCUIT

2

IMAGE PROCESSING TECHNIQUES

2.1

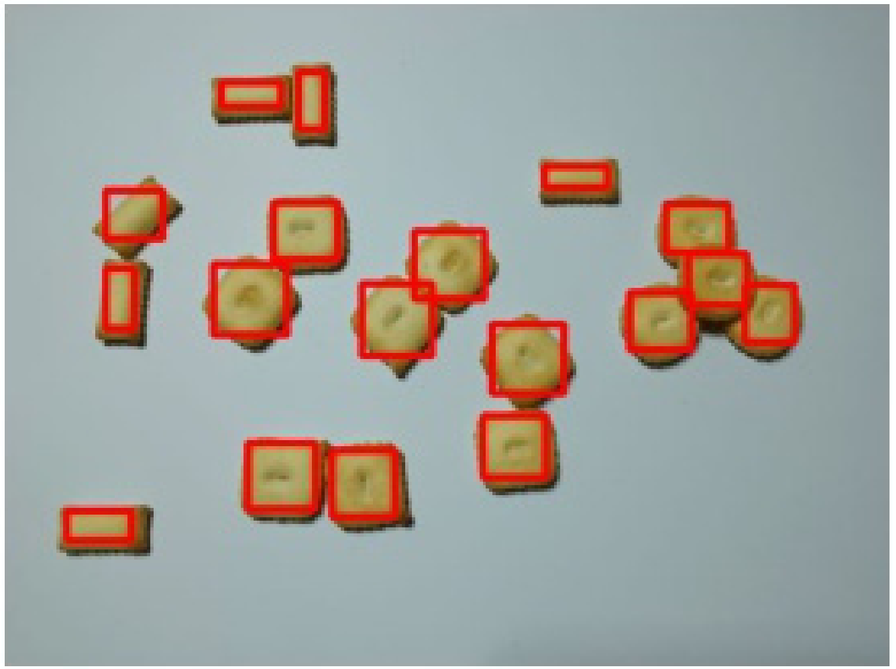
CONNECTED COMPONENT WITH STAT



Component

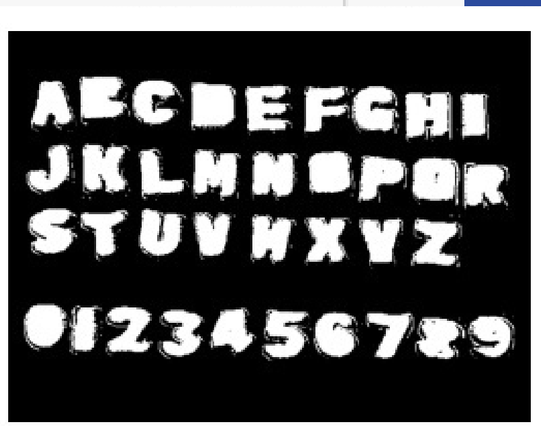
1 เป็นหนึ่งใน MORPHOLOGICAL IMAGE PROCESSING โดยใช้หลักการทางคณิตศาสตร์ เช่น SET THEORY ในการดึงองค์ประกอบและคุณลักษณะต่างๆของรูปภาพที่เราต้องการ

2 มีข้อดีคือสามารถนับจำนวน BISCUIT ในรูปภาพได้แม่นยำ ถ้า BISCUIT อยู่ห่างกันพอสมควร แต่ข้อเสียคือตอนที่แปลงรูปภาพเป็น BINARY IMAGE ต้องทำการไล่หาค่า THRESHOLD ที่ทำให้ผลลัพธ์ออกมาดีที่สุด รวมถึงหากมีการ EROSION อาจทำให้ BISCUIT หายไป



EXAMPLE CODE

```
# HSV threshold to seperate object and background
image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
for i in range(image.shape[0]):
  for j in range(image.shape[1]):
    if(image[i][j][0] <= 57 and image[i][j][1] >= 20 and image[i][j][2] >= 150):
      image[i][j] = np.array([255, 255, 255])
    else:
      image[i][j] = np.array([0, 0, 0])
```



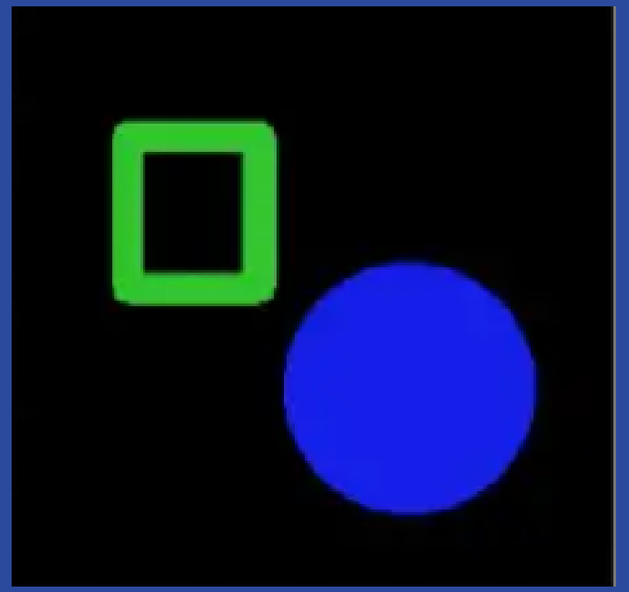
EXAMPLE CODE

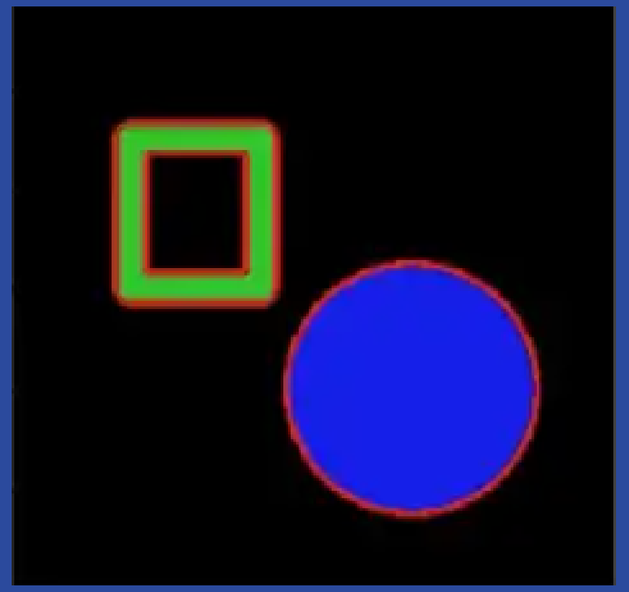
```
kernel = np.ones((8,8),np.uint8)
image = cv2.morphologyEx(image, cv2.MORPH_OPEN, kernel)

kernel = np.ones((5,5), np.uint8)
image = cv2.erode(image, kernel, iterations=1)
```

```
# count
image2 = image[:,:,0]
connectivity = 4
output = cv2.connectedComponentsWithStats(image2, connectivity, cv2.CV_16U)
stats = output[2]

res = []
for i in range(len(stats)):
  if stats[i][0] != 0 and stats[i][4] > 150 and stats[i][4] < 3000 :
    res.append(stats[i])
```

2.2

FIND CONTOURS

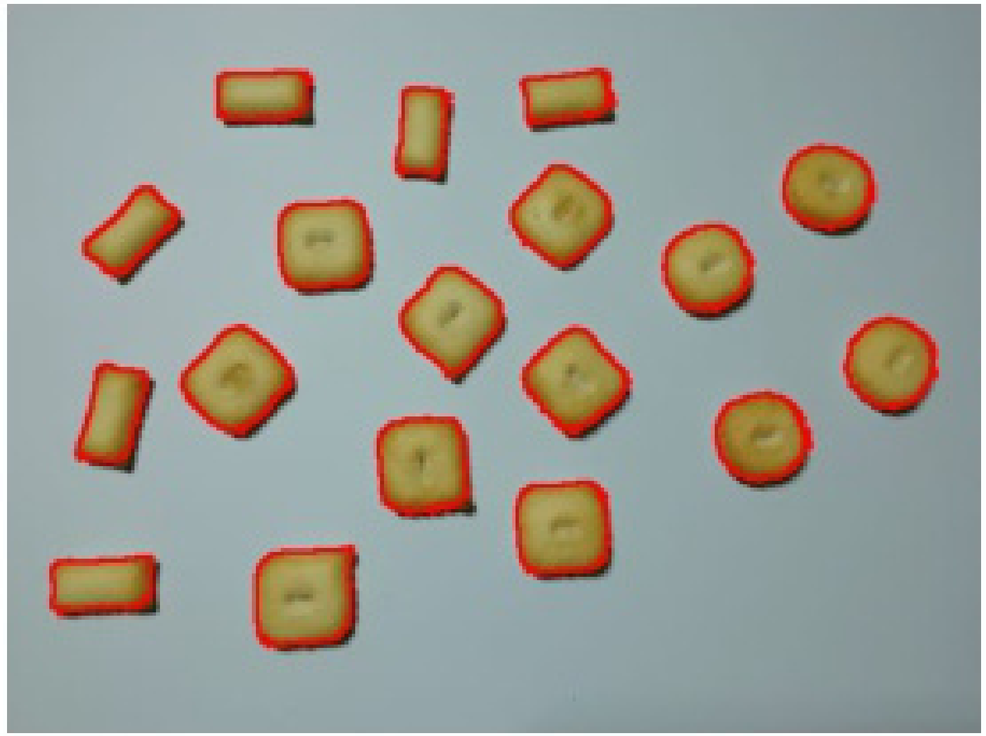




Techniques

1. ใช้ในการหาเส้นขอบของภาพ โดยสามารถใช้คำสั่ง FIND CONTOURS จาก OPEN CV ซึ่งจะ DETECT การเปลี่ยนแปลงของสีในรูปภาพ

2. ข้อดีคือสามารถใช้งาาได้ง่าย แต่มีข้อเสียๆ หลักคือเวลา BISCUIT วางใกล้ชิดกันมากๆ หรือมองซ้อนกันจะ DETECT ได้เพียง BISCUTE 1 ชิ้นเท่านั้น



EXAMPLE CODE

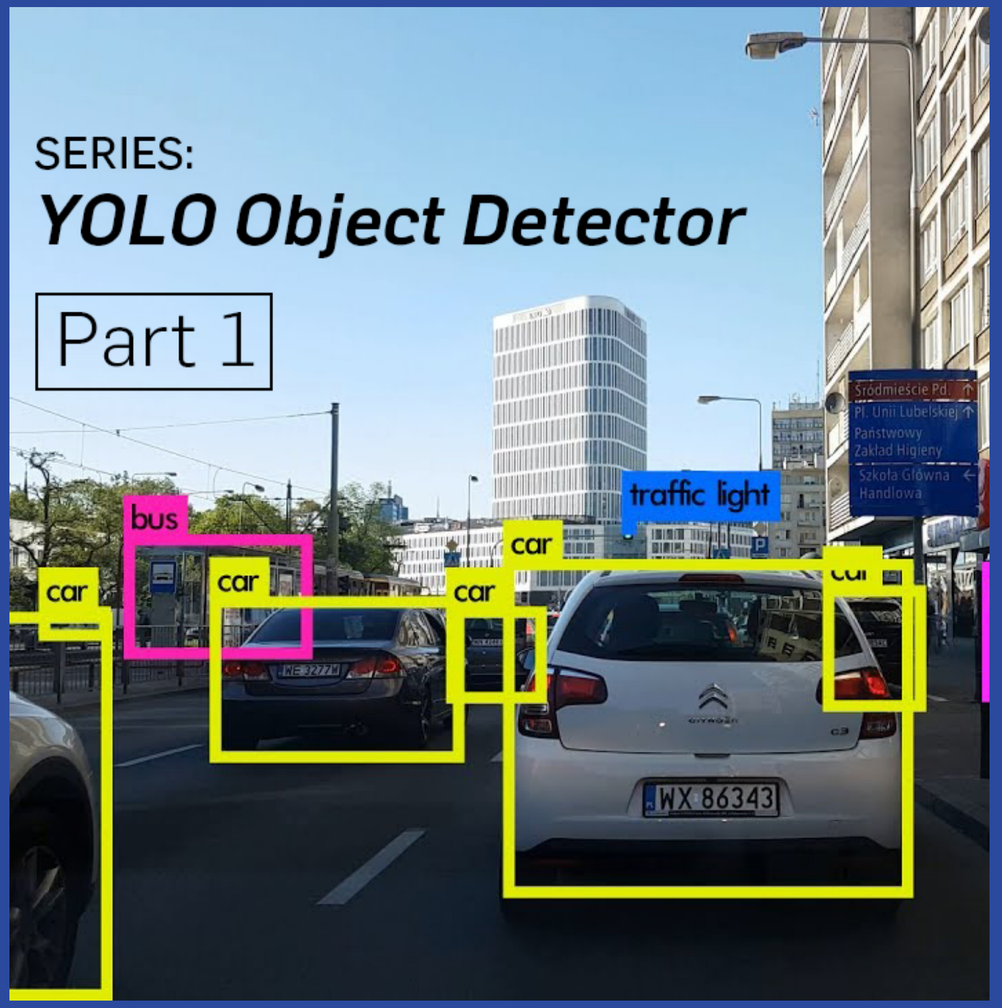
```
# HSV threshold to seperate object and background
image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
for i in range(image.shape[0]):
  for j in range(image.shape[1]):
    if(image[i][j][0] <= 60 and image[i][j][1] >= 60 and image[i][j][2] >= 100):
      image[i][j] = np.array([255, 255, 255])
    else:
      image[i][j] = np.array([0, 0, 0])
ax1.axis('off')
ax1.imshow(image)
ax1.set_title('HSV threshold')
```

EXAMPLE CODE

```
h, s, v1 = cv2.split(image)
canny = cv2.Canny(v1, 100, 200, 3)
canny_show = cv2.Canny(v1, 100, 200, 3)
dilation = cv2.dilate(canny,np.ones((5,5),np.uint8),iterations = 1)
erosion = cv2.erode(dilation,np.ones((3,3),np.uint8),iterations = 1)

(cnt, hierarchy) = cv2.findContours(
    erosion.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

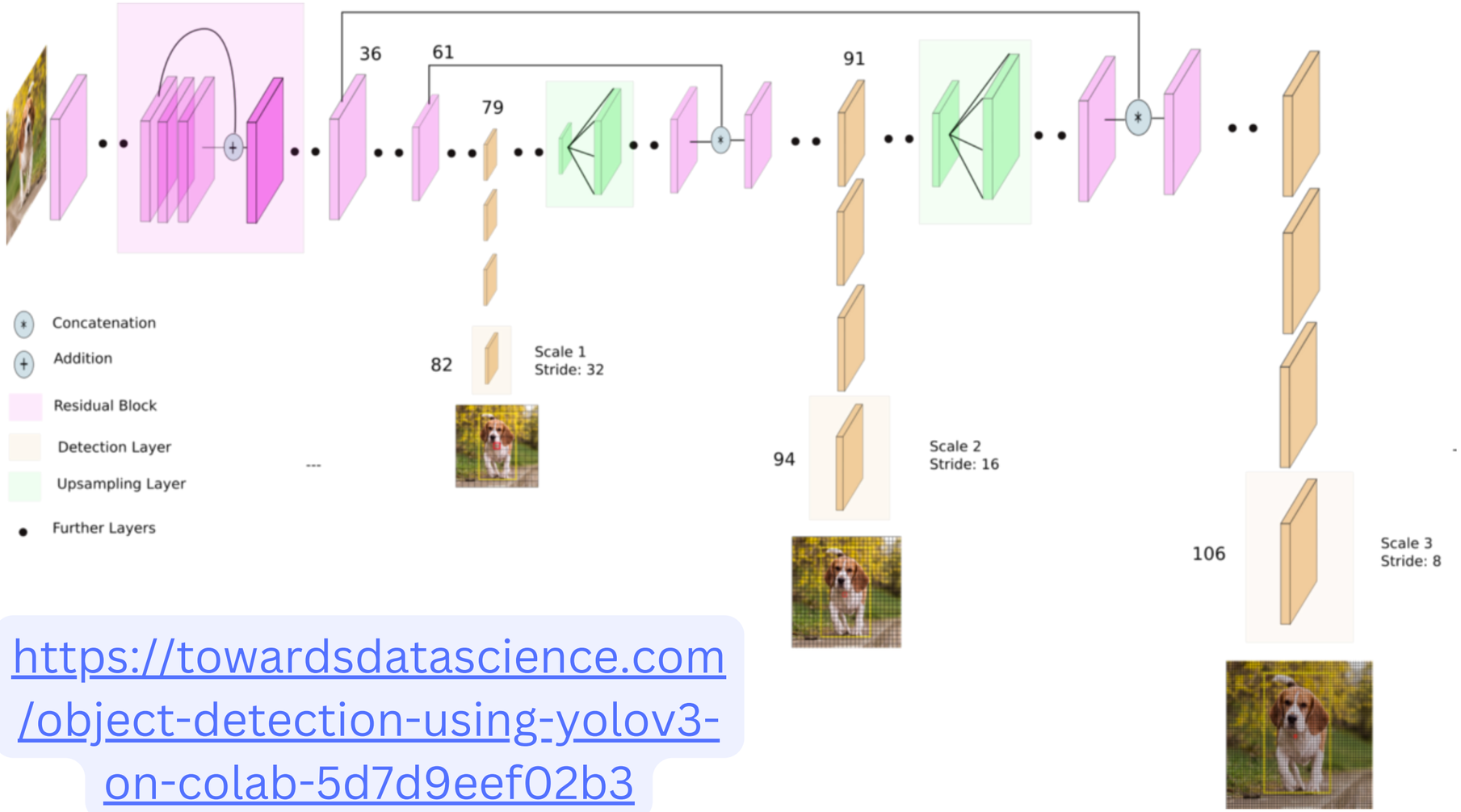
2.3

YOLO v3


SERIES:
YOLO Object Detector
Part 1
bus
car
car
car
car
traffic light
Śródmieście Pd.
Pl. Unii Lubelskiej
Państwowy
Zakład Higieny
Szkoła Główna
Handlowa
WX 86343

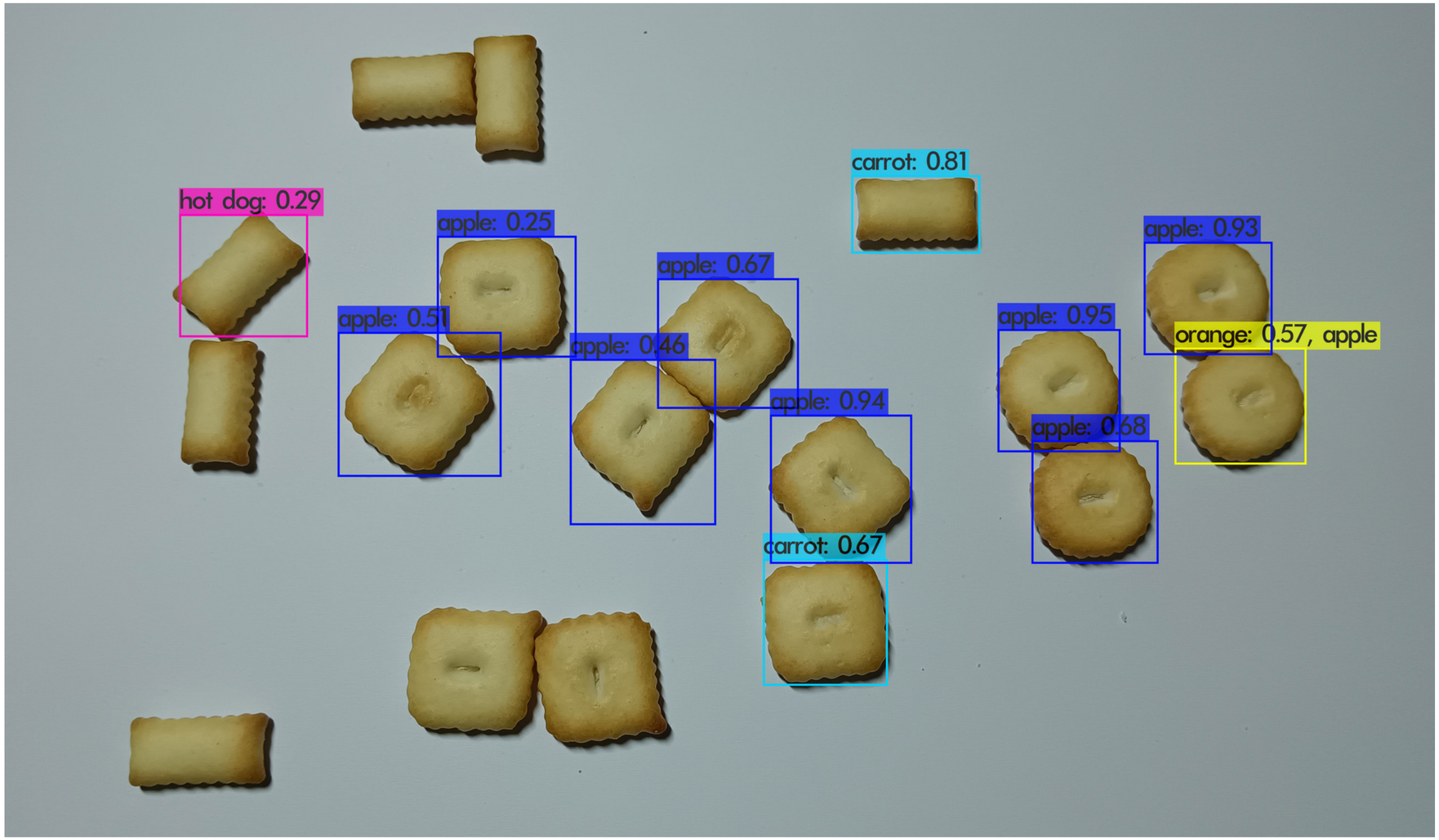
YOLO v3

YOLO is Convolutional Neural Network

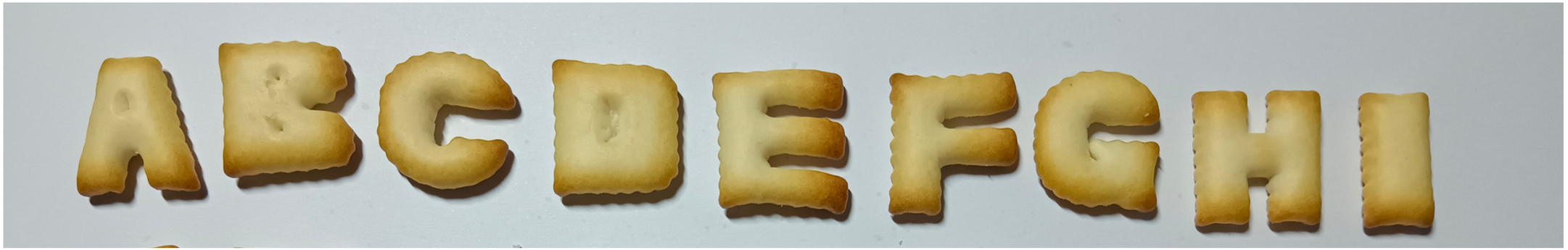


https://towardsdatascience.com/object-detection-using-yolov3-on-colab-5d7d9eef02b3

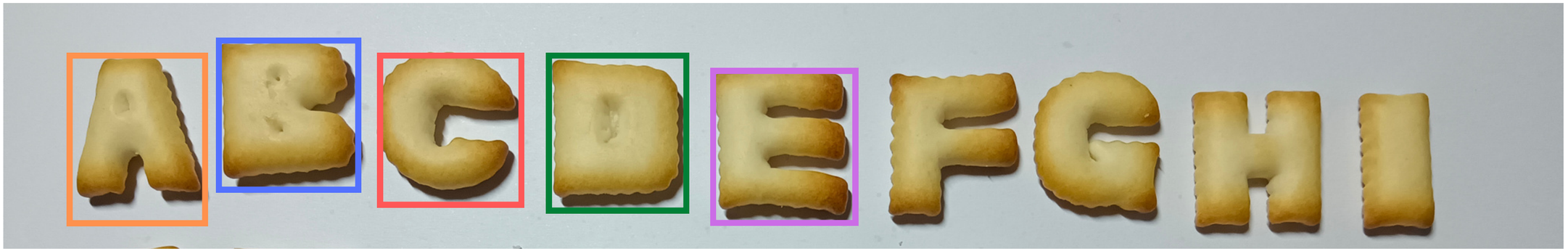
YOLO v3


hot dog: 0.29
apple: 0.25
apple: 0.51
apple: 0.67
apple: 0.46
apple: 0.94
carrot: 0.81
carrot: 0.67
apple: 0.95
apple: 0.93
orange: 0.57, apple
apple: 0.68

YOLO v3



data to train model



class: A

class: B

class: C

class: D

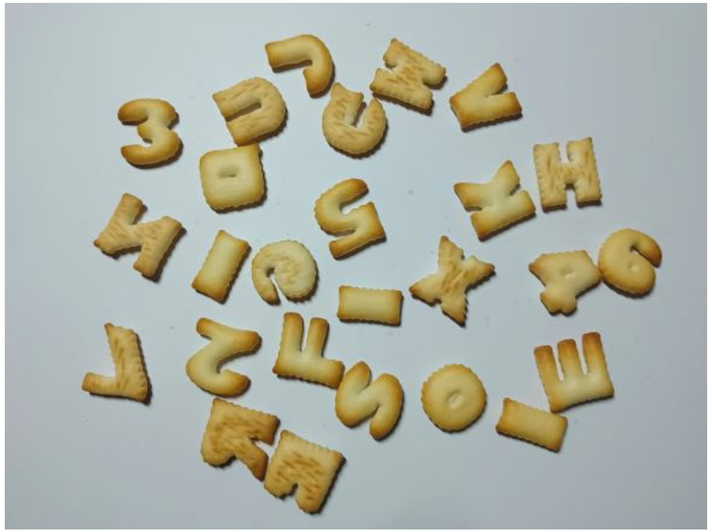
class: E

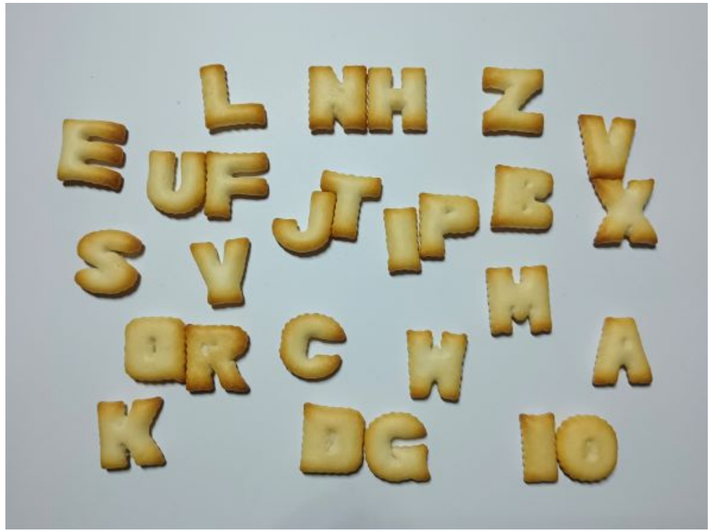
3

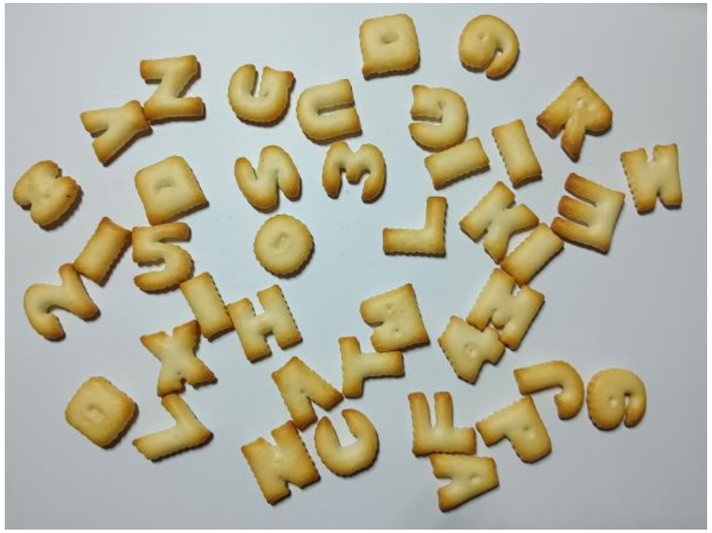
RESULT & TECHNIQUE COMPARE

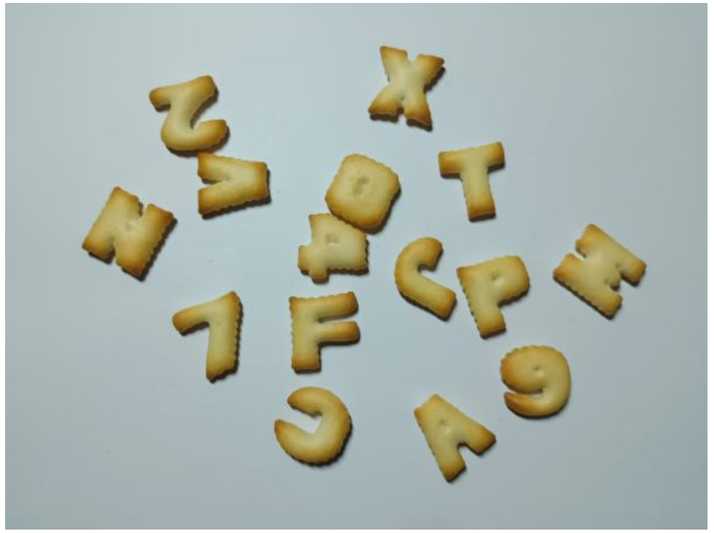
Test Data (Total 31 pictures)

https://drive.google.com/drive/folders/13falNJT_LY8S3t63YfLXbazQMaCTlCuz?usp=share_link





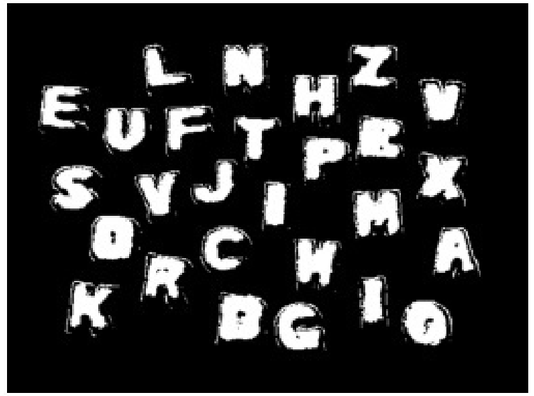




Result: General Case

All result https://drive.google.com/drive/folders/1cLui7lDXmoo1YI0hZNJbr2MrSdnYN9cA?usp=share_link

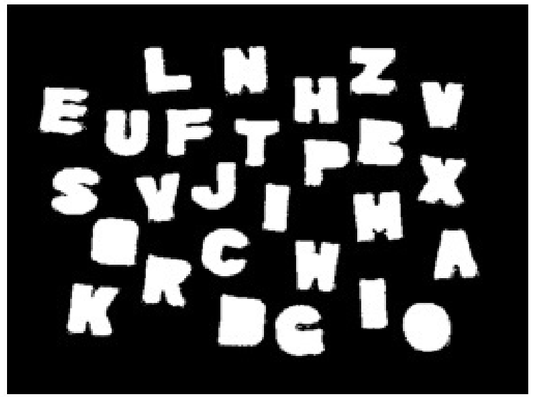
HSV threshold



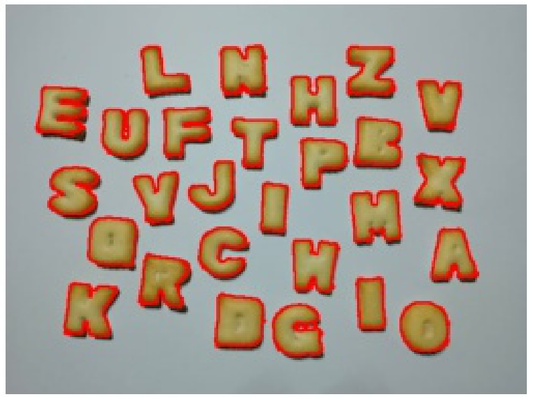
Count by ConnectedComponent

Original

HSV threshold



Count by Contour



Expect: 27

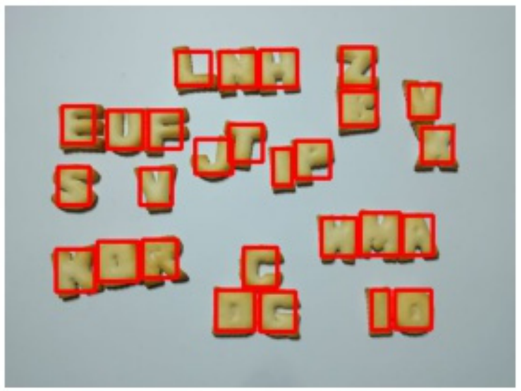
by ConnectedComp: 27

by Contour: 27

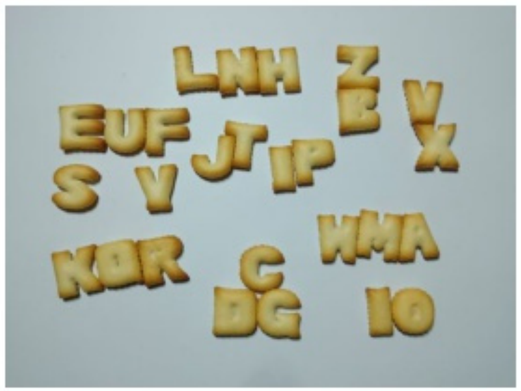
Result: object ชิดกัน

HSV threshold

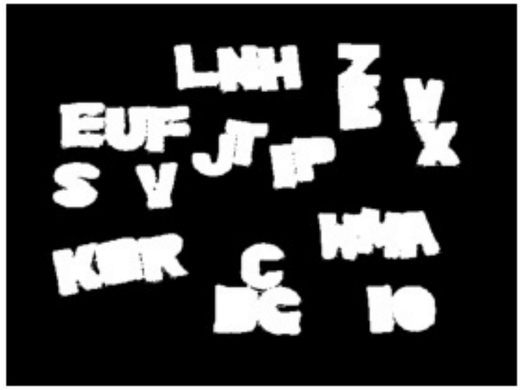
Count by ConnectedComponent



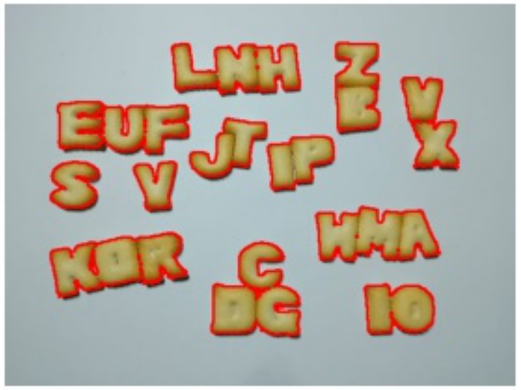
Original



HSV threshold



Count by Contour



Expect: 27

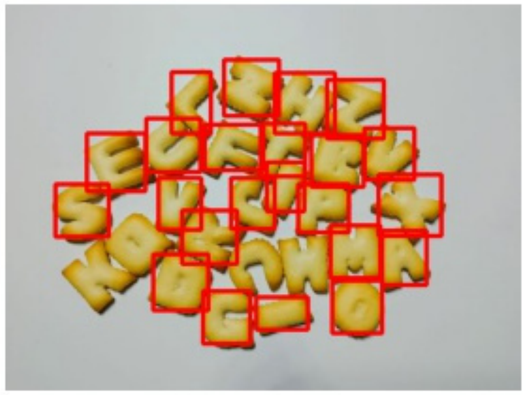
by ConnectedComp: 27

by Contour: 12

Result: object ชิดกันมากๆ

HSV threshold

Count by ConnectedComponent



Original

HSV threshold

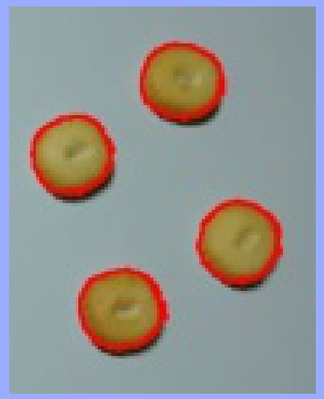
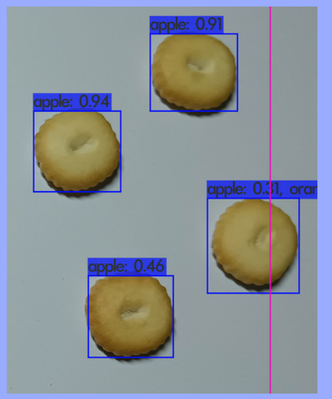
Count by Contour

Expect: 27

by ConnectedComp: 23

by Contour: 1

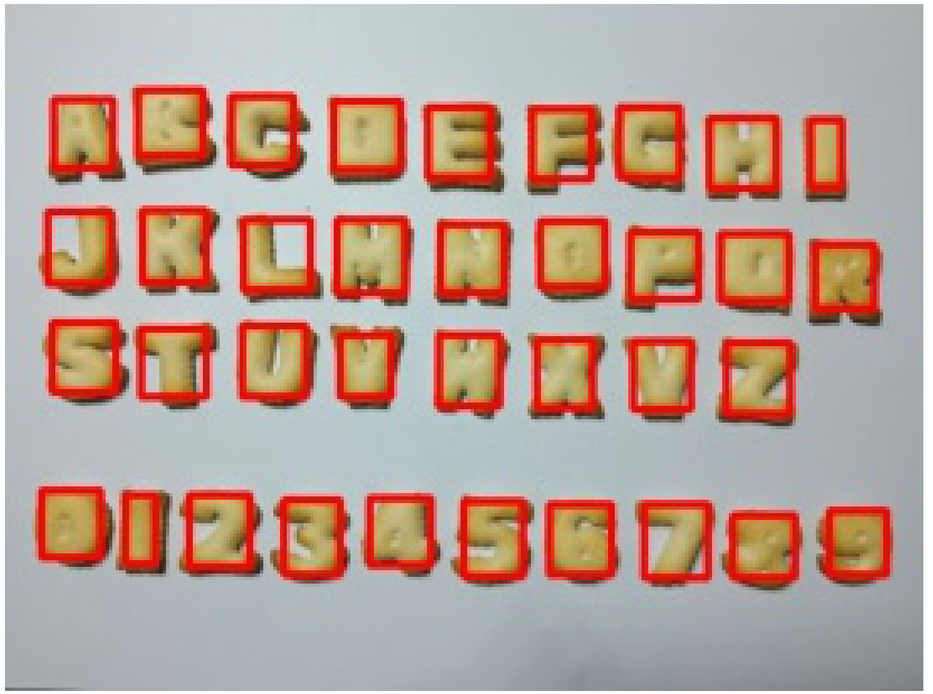
Technique Compare

	Connected component	Contours	YOLO
			
pros	simple to seperating adjacent object	can find the object shape and area	can classify the letter
cons	some object missing after erosion	problem with seperating adjacent object	need a large amount of data to train model
RMSE	1.0000	9.8078	-

Best Technique

CONNECTED COMPONENT (RMSE=1.000)

Expect: 36



Count: 36

THANK YOU

FOR YOUR ATTENTION