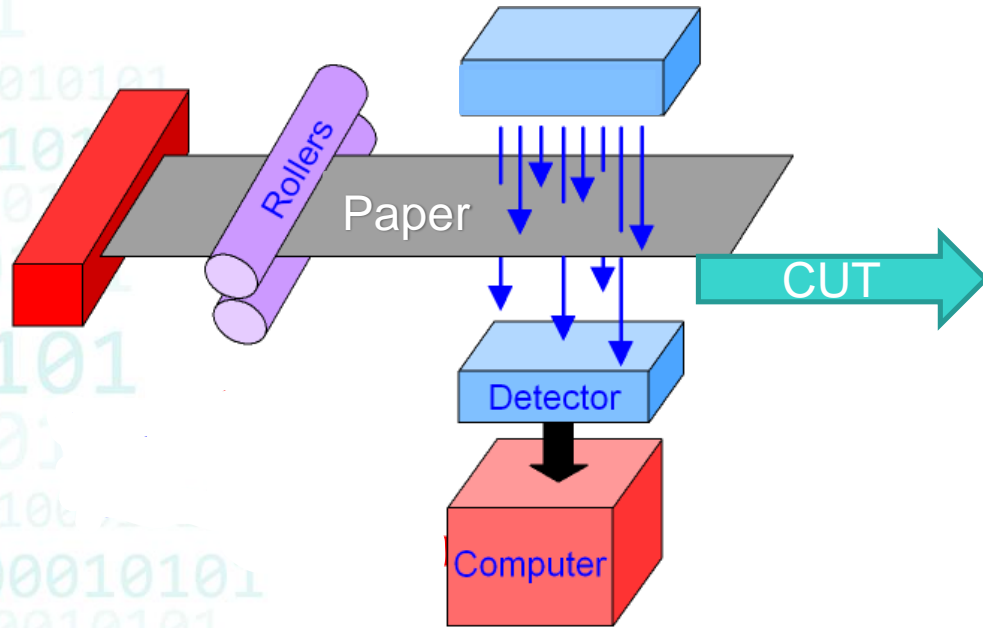
The background features two large, overlapping teal geometric shapes, one in the top-left and one in the bottom-left, both pointing towards the center. The entire background is covered with a faint, light-blue binary code (0s and 1s) pattern.

ระบบตรวจคุณภาพกระดาษด้วย Image Processing

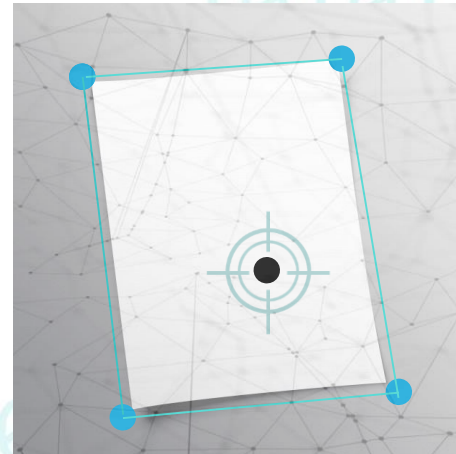
Palapol Chobisara

State of problem

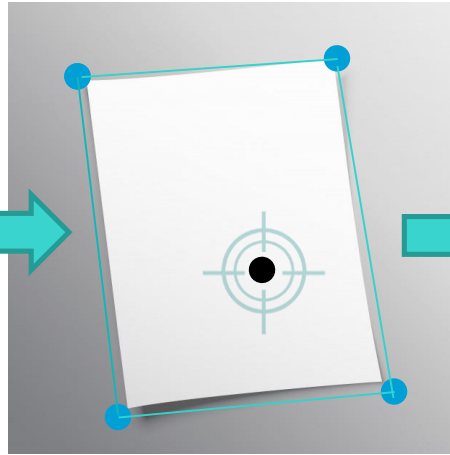


Objective

- เพื่อลดต้นทุนด้านบุคลากร
- เพื่อเพิ่มความแม่นยำในการวัดคุณภาพกระดาษ
- เพื่อเพิ่มความเร็วในการตรวจสอบกระดาษ



Methodology



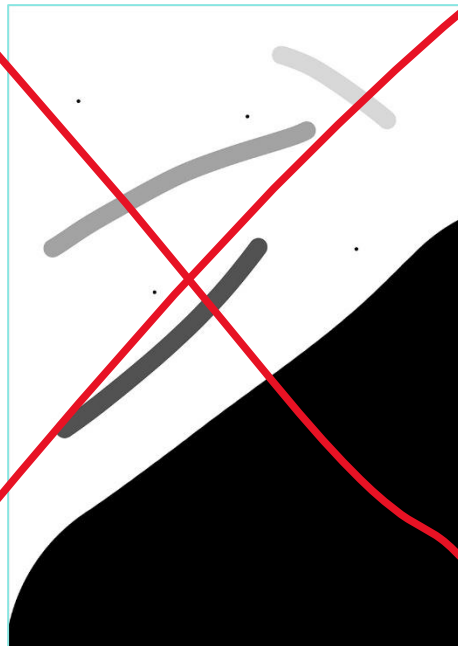
เกรดกระดาษ

Gold
0.01%

• Silver
0.05%

•
Bronze
•
0.1%

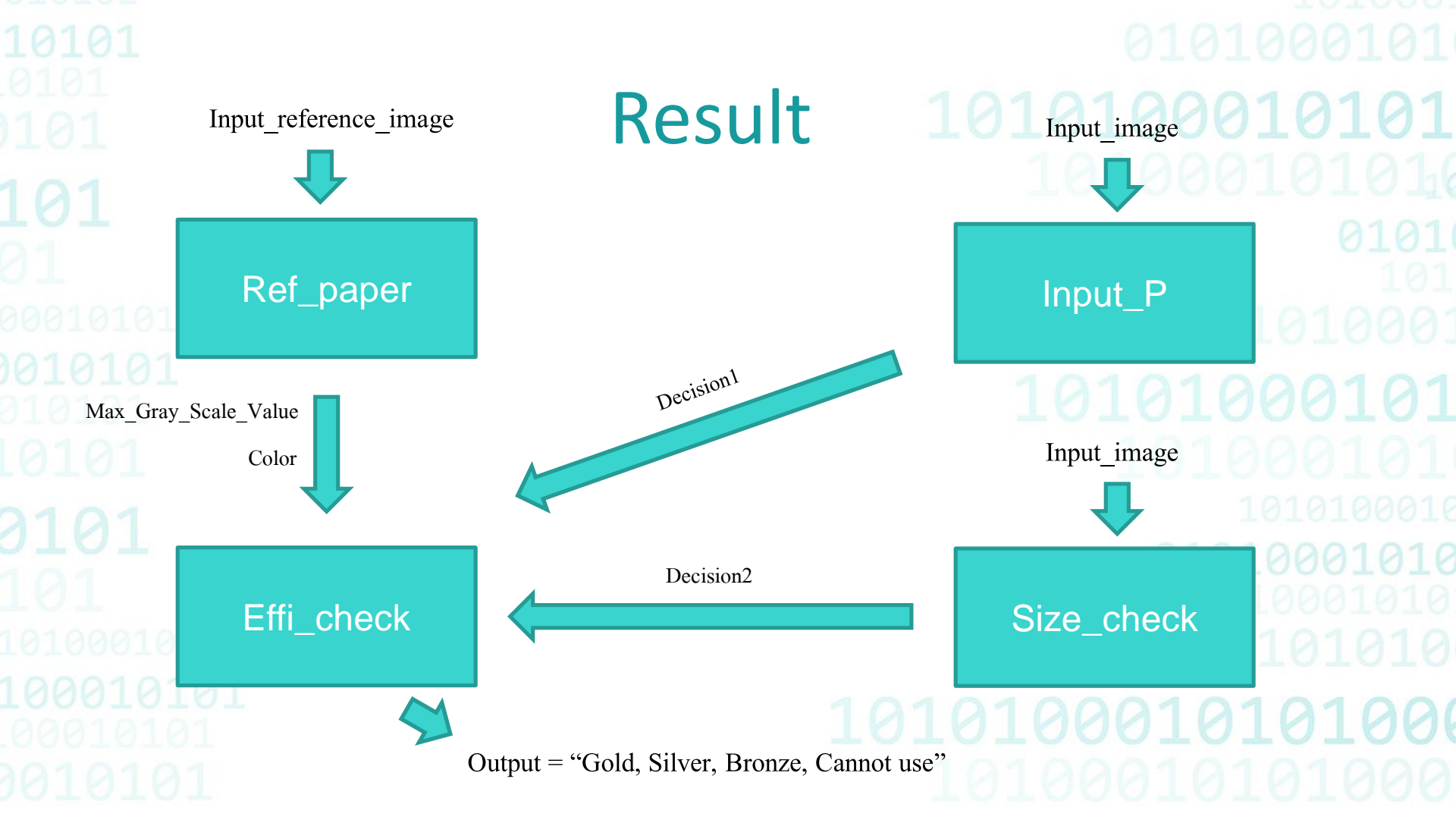
เกรดกระดาษ



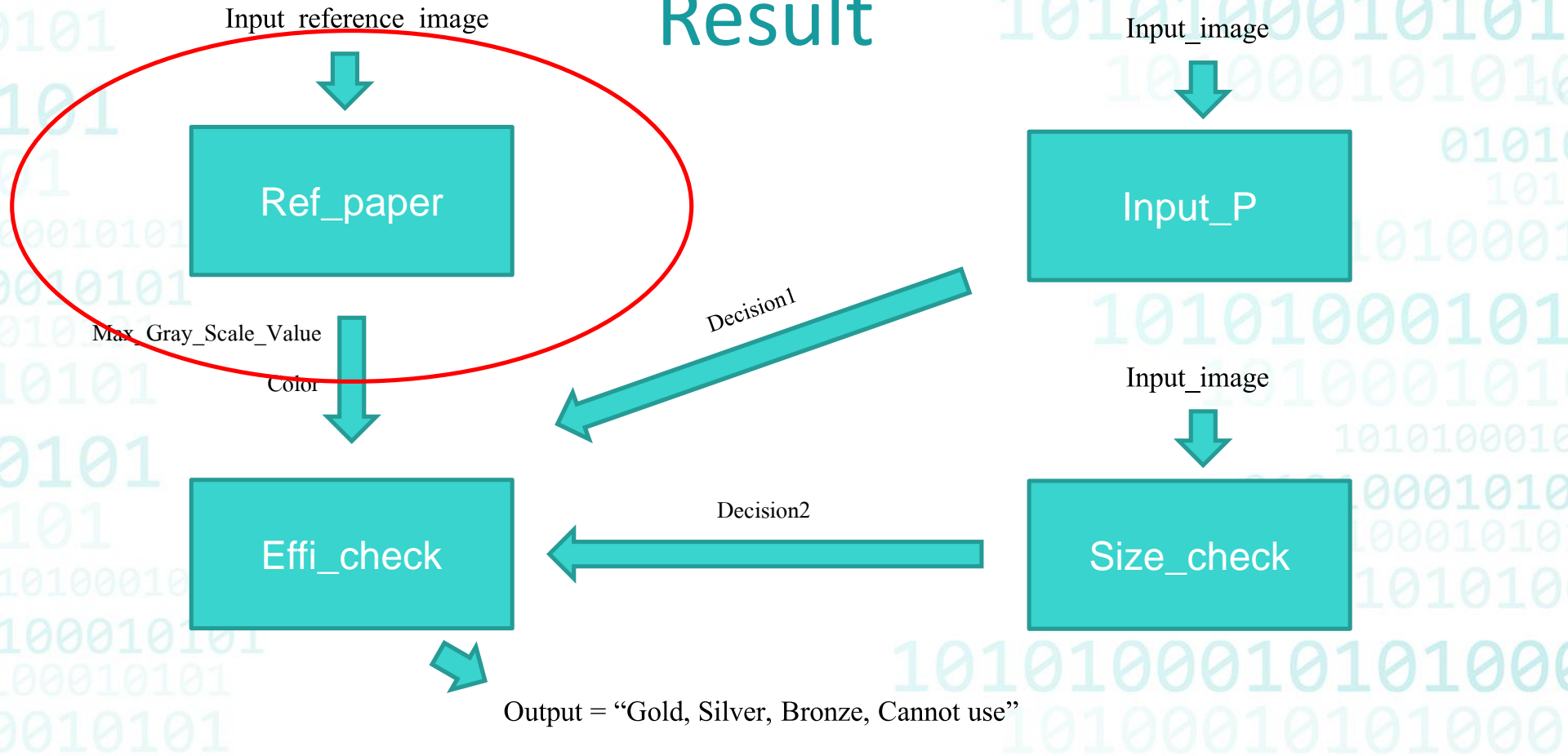
Cannot use

>0.1%

Result



Result

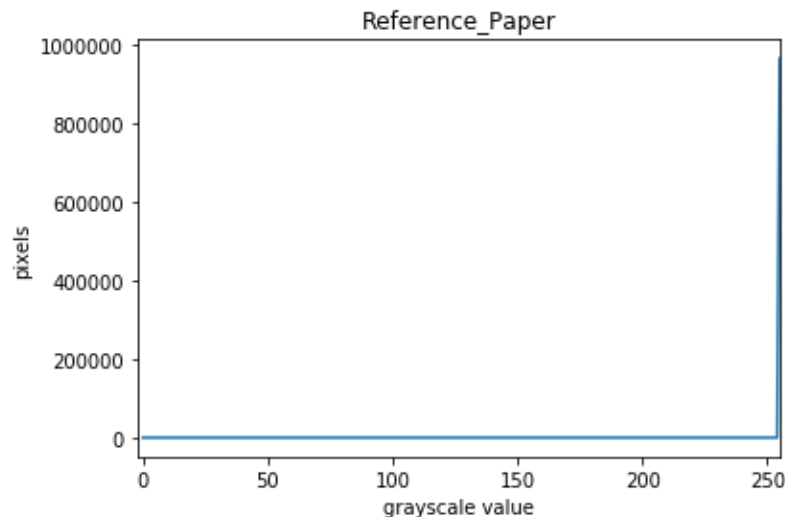


Ref_paper

Ref_input

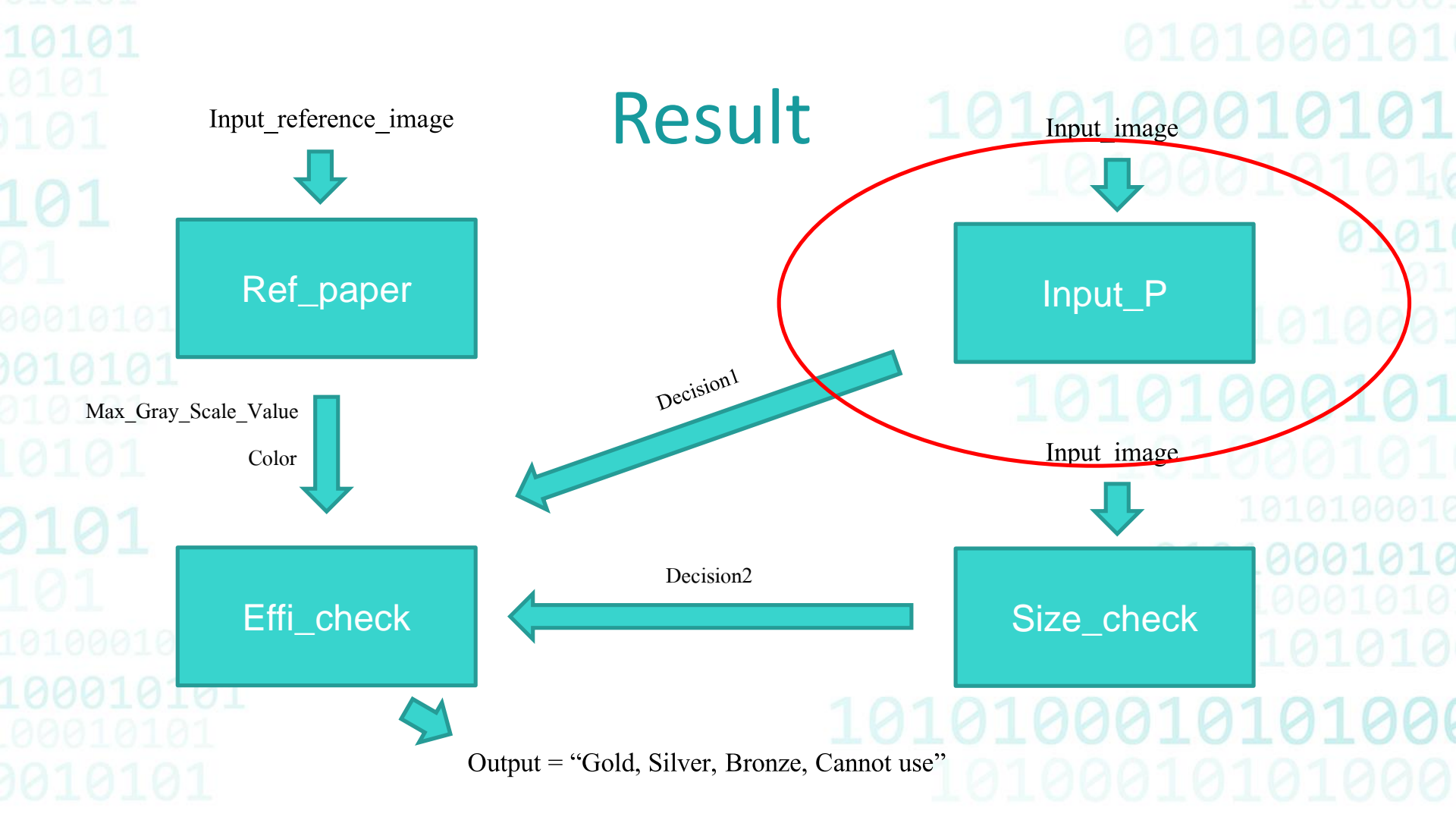
Result

```
def ref_paper(ref_photo):  
    ref_white_num = 0;  
    img = cv2.imread(ref_photo, cv2.IMREAD_COLOR)  
    plt.imshow(img)  
    histogram = cv2.calcHist([img], [0], None, [256], [0, 256])  
    max_value = max(histogram);  
    for j in range(len(histogram)):  
        if max_value == histogram[j]:  
            ref_white_num = max_value;  
            ref_white_num = ref_white_num[0];  
            color = j;  
  
    plt.figure()  
    plt.title("Reference_Paper")  
    plt.xlabel("grayscale value")  
    plt.ylabel("pixels")  
    plt.xlim([-2, 255])  
  
    plt.plot(histogram)  
    plt.show()  
  
    print("White num of ref_paper = ",ref_white_num);  
    print("max histogram is = ",color);  
    return ref_white_num,color;
```

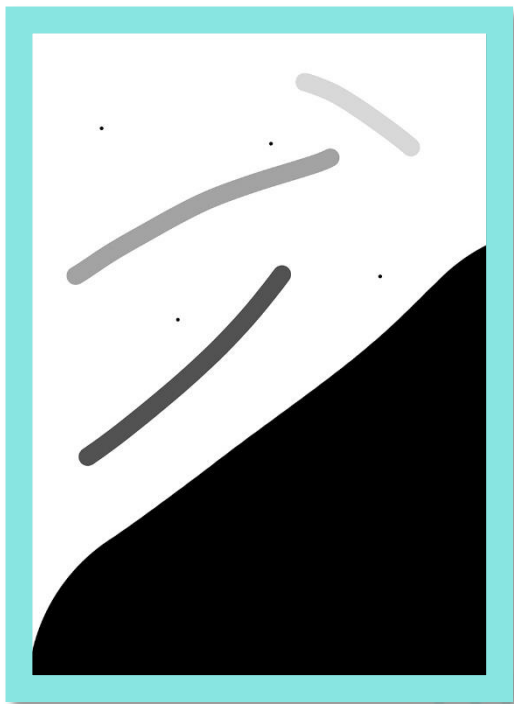


White num of ref_paper = 966763.0
max histogram is = 255

Result

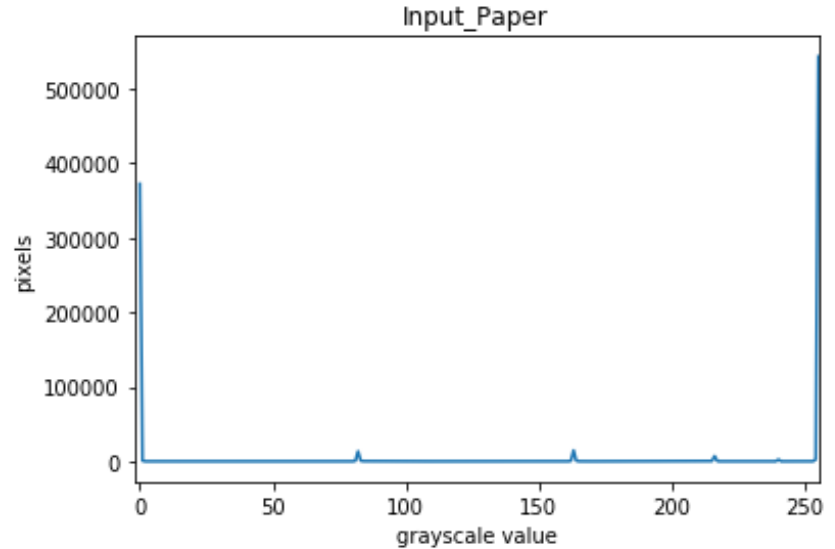


Input_P



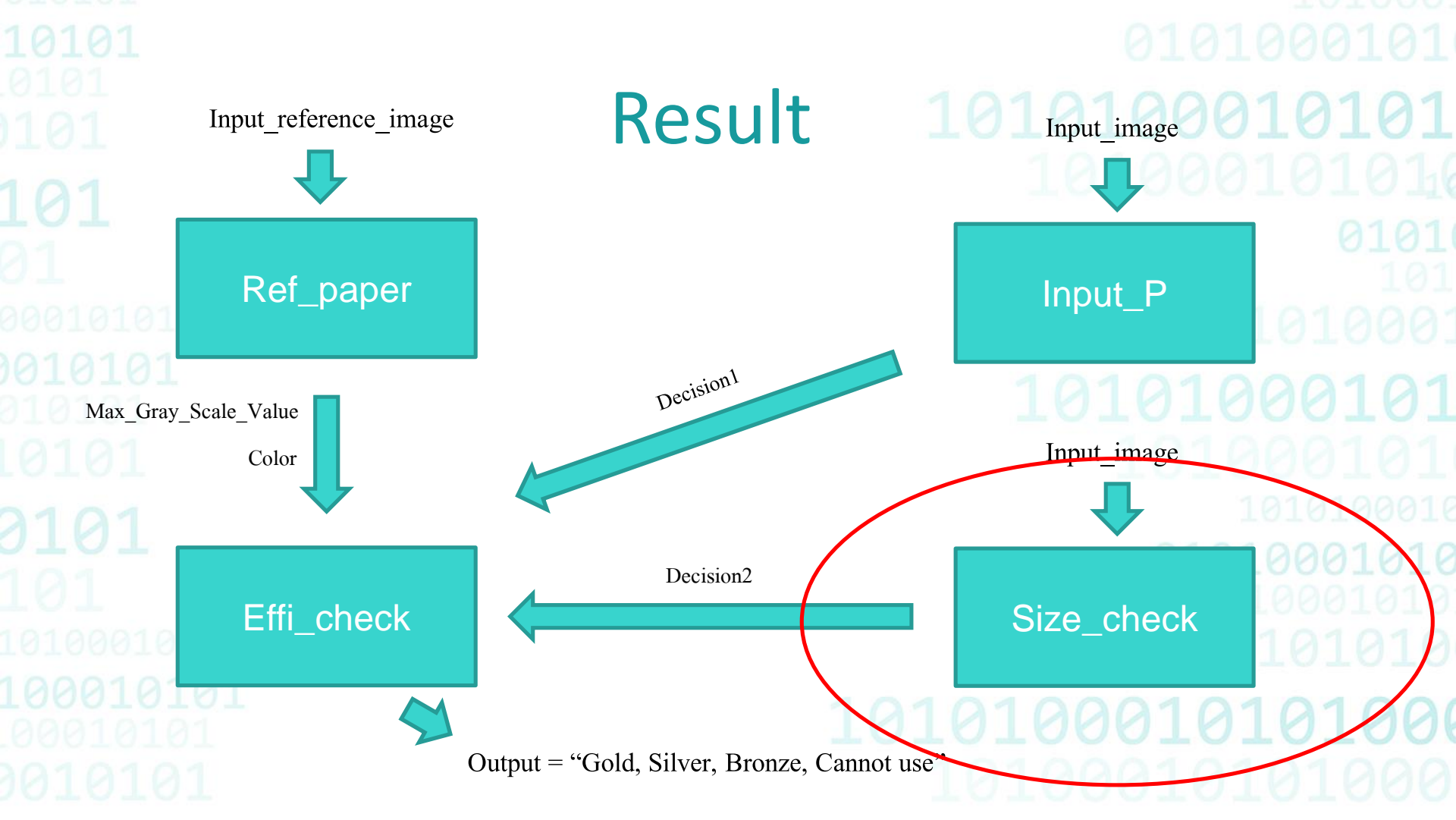
Result

```
def input_P(input_photo):  
    img = cv2.imread(input_photo, cv2.IMREAD_COLOR);  
    #plt.imshow(img)  
    histogram = cv2.calcHist([img], [0], None, [256], [0, 256]);  
    white_paper = 0;  
  
    plt.figure()  
    plt.title("Input_Paper")  
    plt.xlabel("grayscale value")  
    plt.ylabel("pixels")  
    plt.xlim([-2, 255])  
  
    plt.plot(histogram)  
    plt.show()  
  
    for i in range(10):  
        white_paper += histogram[i+246];  
    if white_paper > 700000:  
        decision=1;  
    else:  
        decision=0;  
  
    return decision;
```

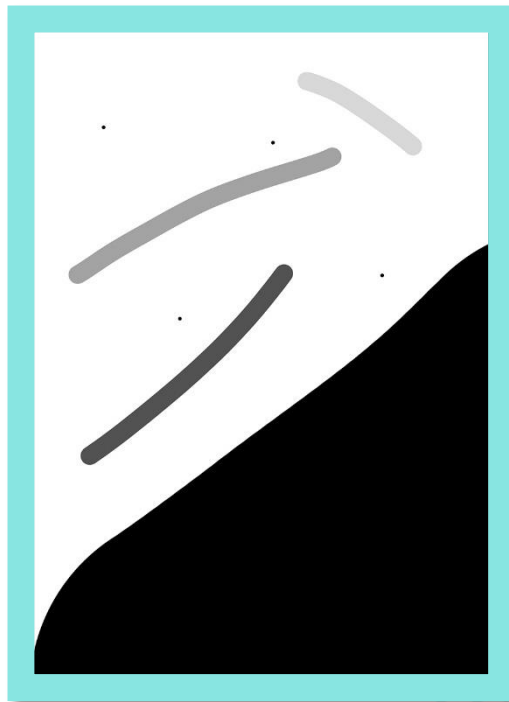


โค้ดเช็คกระดาษ

Result

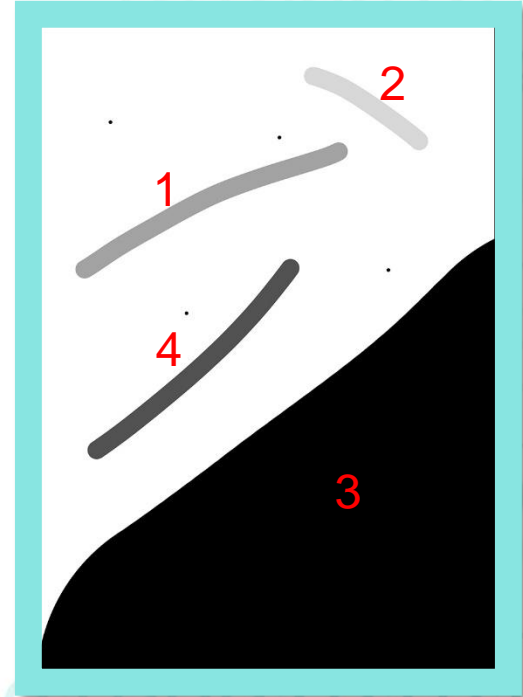


Size_check



Result

```
def size_check(input_photo):  
    decision2=1;  
    img = cv2.imread(input_photo, cv2.IMREAD_COLOR)  
    img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY);  
    ret,thresh = cv2.threshold(img,100,256,0)  
    im2,contours,hierarchy = cv2.findContours(thresh, 1, 2)  
    all_object = [];  
    for item in range(len(contours)):  
        cnt = contours[item]  
        all_object = len(cnt)  
        if all_object >= 5:  
            decision2=0;  
            return decision2;  
        else:  
            decision2=1;  
    return decision2;
```



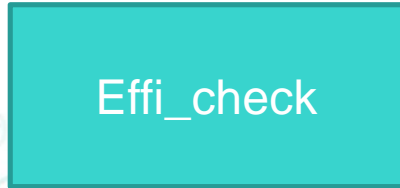
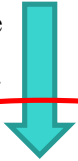
Result

Input_reference_image



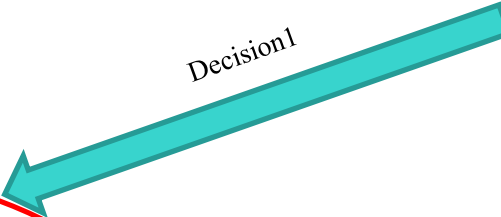
Max_Gray_Scale_Value

Color

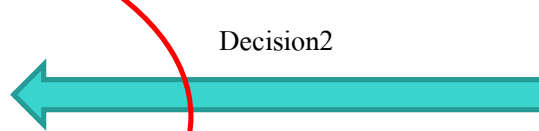


Output = "Gold, Silver, Bronze, Cannot use"

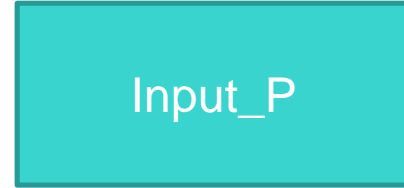
Decision1



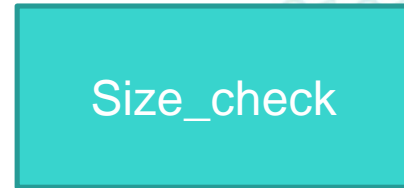
Decision2



Input_image



Input_image



Size_check

```

def effi_check(input_photo,ref_white_num,Percent_effi,decision,color,decision2):
    img = cv2.imread(input_photo,cv2.IMREAD_COLOR)
    histogram = cv2.calcHist([img], [0], None, [256], [0, 256])

    expect_Gold_White_num = ref_white_num-(Percent_effi[0]*(ref_white_num))
    expect_Silver_White_num = ref_white_num-(Percent_effi[1]*(ref_white_num))
    expect_Broze_White_num = ref_white_num-(Percent_effi[2]*(ref_white_num))

    max_value = max(histogram);

    print('Expect for gold   = more than %d' %expect_Gold_White_num)
    print('Expect for silver = more than %d' %expect_Silver_White_num)
    print('Expect for bronze = more than %d' %expect_Broze_White_num)
    print(" ")

    for k in range(len(histogram)):
        if max_value == histogram[k]:
            input_white_num = max_value;
            input_white_num = input_white_num[0];
            color = k;

    if decision == 1 and decision2 == 1:
        print('Input_White_Num = %d' %histogram[color])
        if histogram[color] >= expect_Gold_White_num:
            print("'Gold Grade'");
        elif histogram[color] >= expect_Silver_White_num:
            print("'Silver Grade'");
        elif histogram[color] >= expect_Broze_White_num:
            print("'Bronze Grade'");
        else:
            print("'Can not use'");
    else:
        print(" ")
        print('Input_White_Num = %d' %histogram[color])
        print(" ");
        print("'Can not use'");

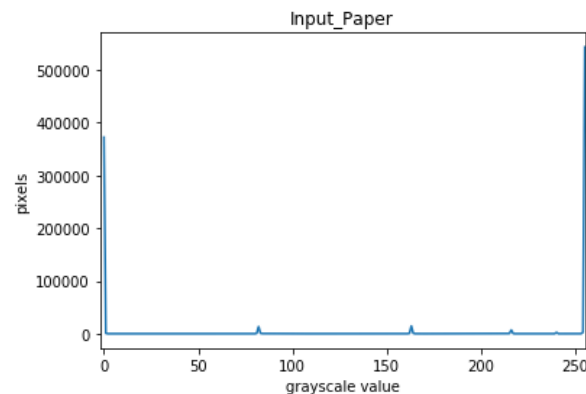
```

Effi_check

เกรดกระดาษที่คาดหวัง

คัดเกรดกระดาษหา

Decision1,2 = 1



Expect for gold = more than 966666
 Expect for silver = more than 966279
 Expect for bronze = more than 965796

Input_White_Num = 542810

'Can not use'

Result

Input	Output
จุดขนาด 3x3 พิกเซล 1 จุด	Gold
จุดขนาด 3x3 พิกเซล 3 จุด	Gold
จุดขนาด 3x3 พิกเซล 6 จุด	Silver
จุดขนาด 3x3 พิกเซล 20 จุด	Cannot use
จุดขนาด 10x10 พิกเซล 1 จุด	Cannot use
ใส่กระดาษที่มีสีเหมือนกระดาษถนอมสายตา	Cannot use
ใส่กระดาษยับที่มีรอยยับจำนวนมาก	Cannot use
ใส่กระดาษยับที่มีรอยยับขนาดเล็ก	Cannot use
ใส่กระดาษสีเทา	Cannot use

วิจารณ์ผลการทดลอง

- กระดาษที่ใช้ในโครงการคือกระดาษที่สร้างมาจากโปรแกรมโฟโต้ช้อป ทำให้ผลลัพธ์ออกมาน่าพอใจ
- ความละเอียดของภาพมีผลกับโครงการนี้เป็นอย่างมาก
- เวลาแสดงฮิสโตแกรม ต้องแสดงตั้งแต่ -2 ถึง 256

สรุปผลการทดลอง

โปรแกรมนี้สามารถคัดกระดาศทั้ง 4 ประเภทที่กล่าวมาได้เร็วและแม่นยำจริง
แต่เป็นการกำหนดให้สภาพแวดล้อมเป็นอุดมคติ (Noise เป็น 0)

ปล.หากนำไปใช้งานกับสภาพการใช้งานจริง สามารถใส่กระดาศใน
สภาพแวดล้อมที่มีการเปลี่ยนแปลงได้ โดยการใช้สภาพแวดล้อมเดียวกันในการ
ตรวจสอบคุณภาพกระดาศ