

Date: 20220910 รหัสนักศึกษา B6321451 ชื่อ-สกุล นางสาวขวัญจิรา พันธุเกตุ

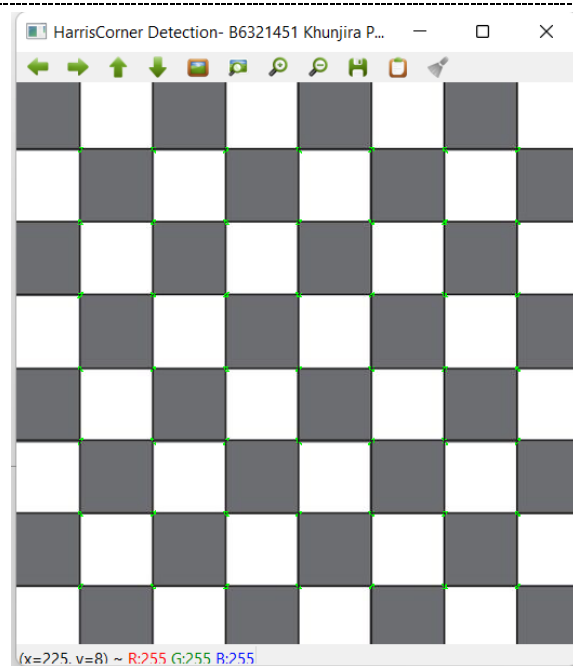
Week05-20220907,20220910 – การเรียนรู้ของเครื่องจักร

- ทำเอกสารส่วนกิจกรรม 1 – กิจกรรม 4 ให้สมบูรณ์
- กำหนดชื่อไฟล์ตามรูปแบบนี้ “B3601234-Week05-นายวิชัย ศรีสุรักษ์.pdf”
- MC ส่งงาน Class Check ก่อน 21:00น วันพุธที่ 07 กย 65 ที่ <https://forms.gle/57M48zaf7pJAPJ5W6>
- MC ส่งงาน Homework ก่อน 06:00น วันพุธที่ 14 กย 65 ที่ <https://forms.gle/nYcZmWuQScyBvFN28>
- PC ส่งงาน Class Check ก่อน 17:00น วันเสาร์ที่ 10 กย 65 ที่ <https://forms.gle/Ff6NEd9gpUdzS1h7A>
- PC ส่งงาน Homework ก่อน 06:00น วันเสาร์ที่ 17 กย 65 ที่ <https://forms.gle/ERvGDuGgajxBV55TA>

Mission1/4:

เปลี่ยนรูปต่อไปนี้ให้เป็นรูปที่มีรหัสและชื่อของเราเอง และเพิ่มรูปอีกสองรูป จาก Google Search

Lab1A - Harris Corner Detection - chessboard



< Capture Code >

```
In [ ]:
# Harris Corner Detection - chessboard
import cv2
import numpy as np
myName = "- B6321451 Khunjira Pantuket"

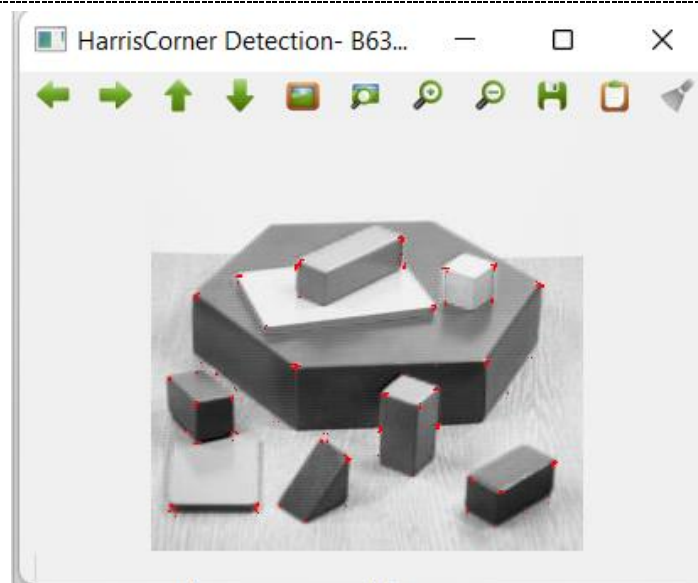
img = cv2.imread('./image/chessboard.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
gray = np.float32(gray)

dst = cv2.cornerHarris(gray, 2, 3, 0.04)

img[dst > 0.01 * dst.max()] = [0, 255, 0]

cv2.imshow("HarrisCorner Detection"+myName, img)
cv2.waitKey()
cv2.destroyAllWindows()
```

Lab1B - Harris Corner Detection - blox



< Capture Code >

```
In [*]:
# Harris Corner Detection - blox
import cv2
import numpy as np
myName = "- B6321451 Khunjira Pantuket"

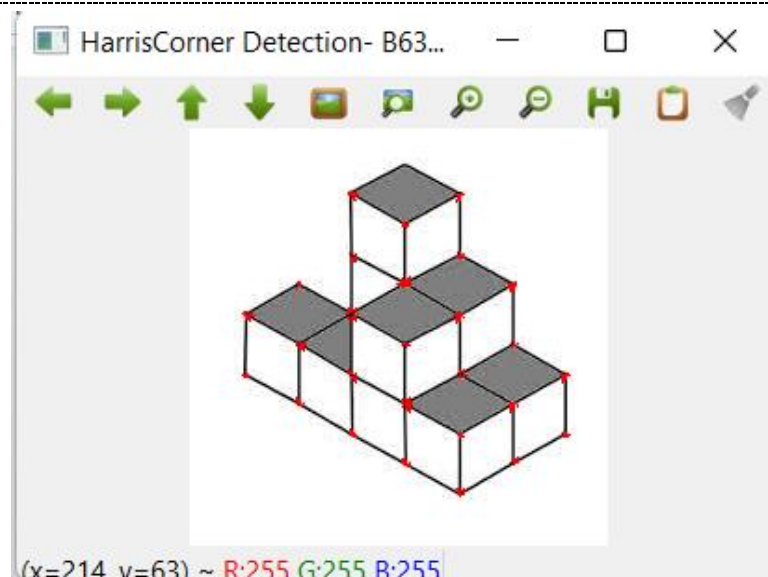
img = cv2.imread('./image/blox.jpg')
gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
gray = np.float32(gray)

dst = cv2.cornerHarris(gray, 2,3,0.04)

img[dst > 0.01 * dst.max()] = [0,0,255]

cv2.imshow("HarrisCorner Detection"+myName,img)
cv2.waitKey()
cv2.destroyAllWindows()
```

Lab1C - Harris Corner Detection – Google-1/4



< Capture Code >

```
In [*]:
# Harris Corner Detection - blox
import cv2
import numpy as np
myName = "- B6321451 Khunjira Pantuket"

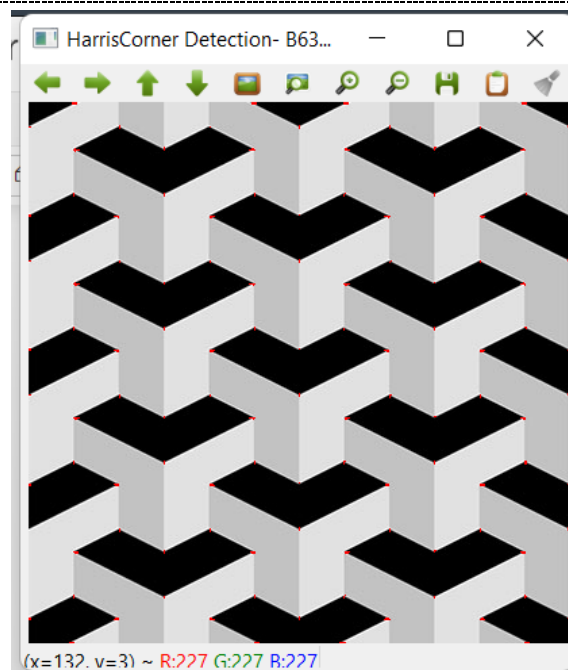
img = cv2.imread("D:\\Machine\\Week5\\me\\3.jpg")
gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
gray = np.float32(gray)

dst = cv2.cornerHarris(gray, 2,3,0.04)

img[dst > 0.01 * dst.max()] = [0,0,255]

cv2.imshow("HarrisCorner Detection"+myName,img)
cv2.waitKey()
cv2.destroyAllWindows()
```

Lab1D - Harris Corner Detection – Google-2/4



< Capture Code >

```
In [*]:
# Harris Corner Detection - blox
import cv2
import numpy as np
myName = "- B6321451 Khunjira Pantuket"

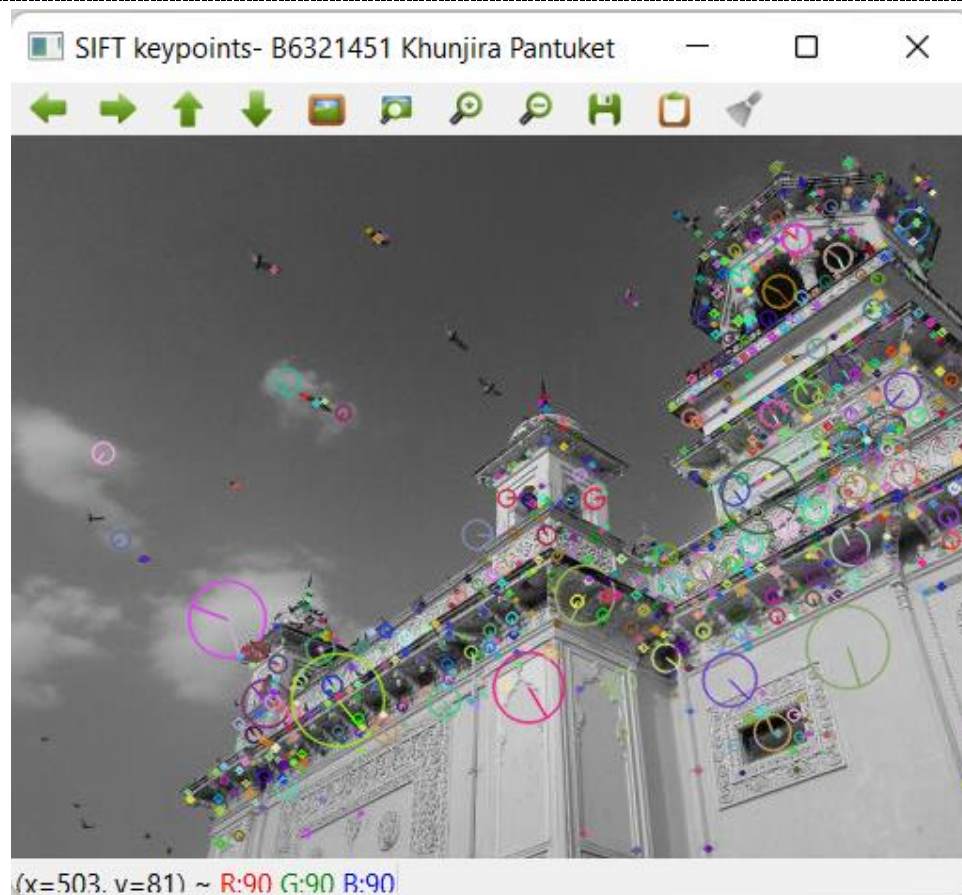
img = cv2.imread("D:\\Machine\\Week5\\me\\6.jpg")
gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
gray = np.float32(gray)

dst = cv2.cornerHarris(gray, 2,3,0.04)

img[dst > 0.01 * dst.max()] = [0,0,255]

cv2.imshow("HarrisCorner Detection"+myName,img)
cv2.waitKey()
cv2.destroyAllWindows()
```

Lab1E - Scale-Invariant Feature Transform -- SIFH



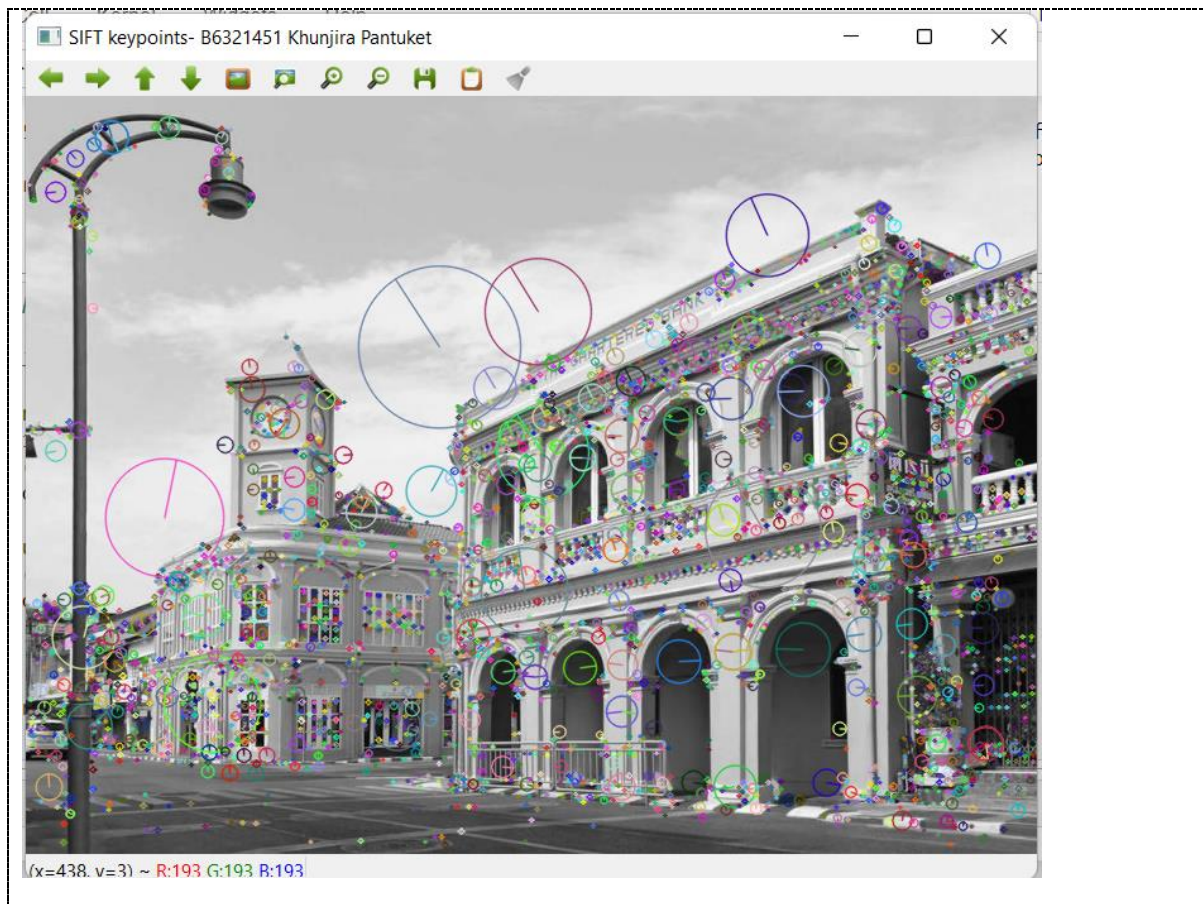
< Capture Code >

```
In [*]:
# SIFT -- Scale-Invariant Feature Transform
import cv2
import numpy as np
myName = "- B6321451 Khunjira Pantuket"
img = cv2.imread('./image/house.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

sift = cv2.xfeatures2d.SIFT_create()
kp = sift.detect(gray, None)
img = cv2.drawKeypoints(gray, kp, img, flags=cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINT)

cv2.imshow('SIFT keypoints'+myName, img)
cv2.waitKey()
cv2.destroyAllWindows()
```

Lab1F - Scale-Invariant Feature Transform – SIFT ใช้รูป Google-3/4



< Capture Code >

```
In [*]: # "D:\Machine\Week5\me\house_me.jpg"

# SIFT -- Scale-Invariant Feature Transform
import cv2
import numpy as np
myName = "- B6321451 Khunjira Pantuket"
img = cv2.imread("D:\\Machine\\Week5\\me\\house_me.jpg")
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

sift = cv2.xfeatures2d.SIFT_create()
kp = sift.detect(gray, None)
img = cv2.drawKeypoints(gray, kp, img, flags=cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)

cv2.imshow('SIFT keypoints'+myName, img)
cv2.waitKey()
cv2.destroyAllWindows()
```


Lab1G - Speeded-Up Robust Features -- SUFT

ข้าม

< Capture Code >


Lab1H – Match Image

Input- B6321451 Khunjira Pantuket




(x=1092, y=79) ~ R:255 G:140 B:255

Ref.- B6321451 Khunjira Pant...



Result- B6321451 Khunjira Pantuket



(x=1119, y=83) ~ R:255 G:255 B:255

< Capture Code >

```

In [*]: import cv2
import numpy as np
# from matplotlib import pyplot as plt

myName = "- B6321451 Khunjira Pantuket"
img_rgb = cv2.imread('./image/FramePemon.png')
template = cv2.imread('./image/template.png')
cv2.imshow("Input"+myName, img_rgb)
cv2.imshow("Ref."+myName, template)
template = cv2.imread('./image/template.png',0)
img_gray = cv2.cvtColor(img_rgb, cv2.COLOR_BGR2GRAY)
w, h = template.shape[::-1]

res = cv2.matchTemplate(img_gray, template, cv2.TM_CCOEFF_NORMED)
threshold = 0.8
loc = np.where(res >= threshold)
print(len(loc))

for pt in zip(*loc[::-1]):
    cv2.rectangle(img_rgb, pt, (pt[0] + w, pt[1] + h), (0, 255, 0), 2)
    print("rectangle 1")

# cv2.imwrite('res.png',img_rgb)
cv2.imshow("Result"+myName, img_rgb)
cv2.waitKey()
cv2.destroyAllWindows()

2
rectangle 1

```

Mission 2/4: กิจกรรม 1

จงสร้างแบบจำลองการถดถอยเชิงเส้นอย่างง่ายสำหรับใช้ในการพยากรณ์ค่าเช่าต่อเดือน (บาท) จากขนาดของพื้นที่ (ตารางเมตร) โดยมีข้อมูลดังต่อไปนี้ไฟล์ Area_Rental.csv

โดยอยากทราบว่าพื้นที่ขนาด 50 ตารางเมตร จะต้องจ่ายค่าเช่าประมาณเดือนละเท่าไร ?

นำเข้า Library ต่างๆ

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%config InlineBackend.figure_format = 'retina'
```

อ่านข้อมูล CSV ไฟล์

```
In [2]: bp = pd.read_csv('./data/Area_Rental.csv')
print(bp)
```

	Area	Rental
0	22.0	2000
1	23.5	2900
2	25.0	3200
3	26.5	3600
4	30.0	3800
5	32.0	4200
6	34.0	4600
7	36.5	5100
8	38.0	5700
9	42.0	6000

ดูค่าสหสัมพันธ์(Correlation) ระหว่าง Mother Blood Pressure กับ Baby Blood Pressure

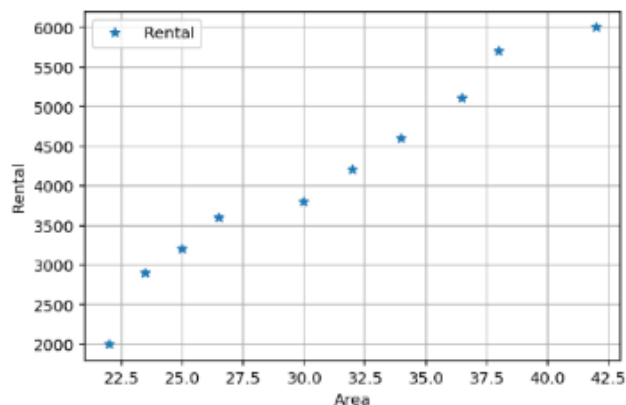
```
In [3]: bp.corr()
```

Out[3]:

	Area	Rental
Area	1.000000	0.982158
Rental	0.982158	1.000000

สร้างแผนภาพการกระจาย (scatter plot) ระหว่าง mother blood pressure และ baby blood pressure

```
In [4]: bp.plot(x='Area', y='Rental', style='*')
plt.xlabel('Area')
plt.ylabel('Rental')
plt.grid()
plt.show()
```



นำเข้า Linear Regression

```
In [5]: from sklearn.linear_model import LinearRegression
```

กำหนดตัวแปรต้น (X) และตัวแปรตาม (y)

```
In [6]: xx = bp[['Area']]
        yy = bp[['Rental']]
```

สร้างและฝึกฝนแบบจำลอง

```
In [7]: lrm = LinearRegression()
        lrm.fit(xx,yy)
```

```
Out[7]: ▼ LinearRegression
        LinearRegression()
```

ดูค่าจุดตัดแกน y (ค่า c)

```
In [8]: lrm.intercept_
```

```
Out[8]: -1629.5818148125263
```

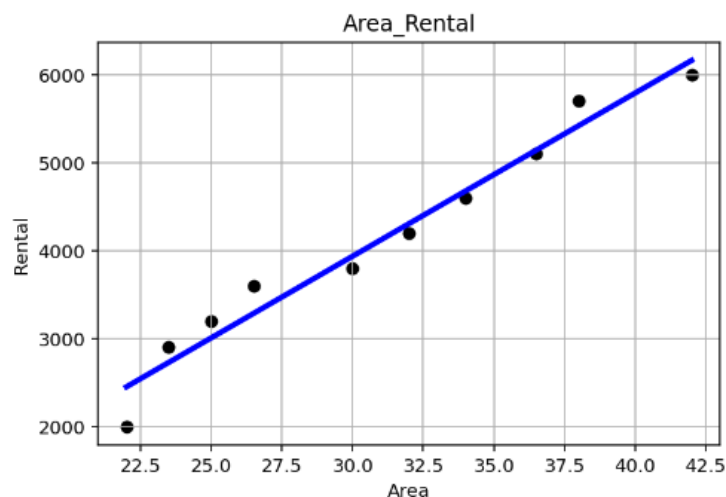
ดูค่าสัมประสิทธิ์ (ค่า m)

```
In [9]: lrm.coef_
```

```
Out[9]: array([185.44690839])
```

การพยากรณ์ด้วยค่า X และสร้างกราฟผลการพยากรณ์

```
In [10]: predictions = lrm.predict(xx)
         plt.scatter(xx, yy, color='black')
         plt.plot(xx, predictions, color='blue', linewidth=3)
         plt.title('Area_Rental')
         plt.xlabel('Area')
         plt.ylabel('Rental')
         plt.grid()
         plt.show()
```



ทดลองพยากรณ์ ข้อมูลใหม่

In [11]: `lrm.predict([[50]])`

C:\Users\ADMIN\miniconda3\lib\site-packages\sklearn
gression was fitted with feature names
warnings.warn(

Out[11]: `array([7642.76360492])`

In [12]: `lrm.predict([[160]])`

C:\Users\ADMIN\miniconda3\lib\site-packages\sklearn
gression was fitted with feature names
warnings.warn(

Out[12]: `array([28041.92352832])`

การประเมินประสิทธิภาพของแบบจำลอง

In [13]: `from sklearn import metrics
print('MAE : ',metrics.mean_absolute_error(yy,predictions))
print('MSE : ',metrics.mean_squared_error(yy,predictions))
print('RMSE: ',np.sqrt(metrics.mean_squared_error(yy,predictions)))`

MAE : 192.56532213231193

MSE : 51028.22904441289

RMSE: 225.89428732133288

Step2/4: การนำเข้าข้อมูล

```
In [4]: train = np.array([])
for i in range(6):
    for j in range(1, 251):
        img = cv2.imread('./image/digits_dataset/'+str(i)+'_'+str(j)+'.png')
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        train = np.append(train, gray)
train = train.reshape(-1,400).astype(np. float32)
print('Get Data for train - Ok')

test = np.array([])
for i in range(6):
    for j in range(251, 501):
        img = cv2.imread('./image/digits_dataset/' +str(i)+'_'+str(j)+'.png')
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        test = np.append(test, gray)
test = test.reshape (-1, 400).astype (np. float32)
print('Get Data for test - Ok')
```

Get Data for train - Ok
Get Data for test - Ok

Step3/4: การสร้างแบบจำลอง

```
In [5]: k = np.arange(6)
train_labels = np.repeat(k, 250)[: ,np.newaxis]
test_labels = train_labels.copy ()

knn = cv2.ml.KNearest_create()
knn.train(train, cv2.ml.ROW_SAMPLE, train_labels)
ret, result, neighbours, dist = knn.findNearest(test, k=5)
result.shape, result
```

```
Out[5]: ((1500, 1),
array([[0.],
       [0.],
       [0.],
       ...,
       [5.],
       [5.],
       [5.]], dtype=float32))
```

Step4/4: การทดสอบแบบจำลอง - ตรวจสอบ Accuracy

```
In [7]: matches = result == test_labels
correct = np.count_nonzero(matches)
accuracy = correct * 100.0 / result.size
print('accuracy = ', accuracy)
```

accuracy = 95.86666666666666

Step4/4: การทดสอบแบบจำลอง – ตรวจสอบด้วย Unknow X

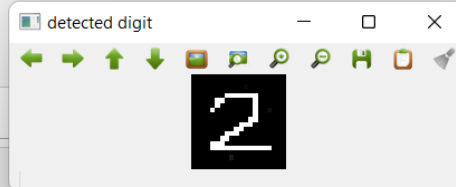
```
In [*]: mydigit = cv2.imread('./image/unknown_X.png')
cv2.imshow('detected digit', mydigit)
mydigit_gray = cv2.cvtColor(mydigit, cv2.COLOR_BGR2GRAY)
mydigit_test = mydigit_gray.reshape((-1,400)).astype(np.float32)
ret, result, neighbours, dist = knn.findNearest(mydigit_test, k=5)
print(ret)

font = cv2.FONT_HERSHEY_SIMPLEX
cv2.putText(mydigit, str(int(ret)), (0, 7), font, 0.3, (255, 0, 0), 1, cv2.LINE_AA)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

2.0

In []:



Step4/4: การทดสอบแบบจำลอง – ตรวจสอบด้วย Unknow Y

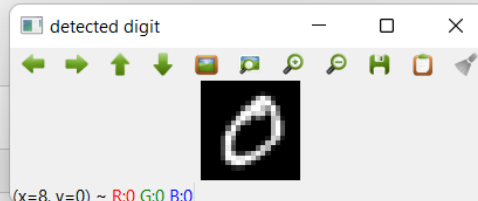
```
In [*]: mydigit = cv2.imread('./image/unknown_Y.png')
cv2.imshow('detected digit', mydigit)
mydigit_gray = cv2.cvtColor(mydigit, cv2.COLOR_BGR2GRAY)
mydigit_test = mydigit_gray.reshape((-1,400)).astype(np.float32)
ret, result, neighbours, dist = knn.findNearest(mydigit_test, k=5)
print(ret)

font = cv2.FONT_HERSHEY_SIMPLEX
cv2.putText(mydigit, str(int(ret)), (0, 7), font, 0.3, (255, 0, 0), 1, cv2.LINE_AA)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

0.0

In []:



Step4/4: การทดสอบแบบจำลอง – ตรวจสอบด้วย Unknow Z

```
In [*]: mydigit = cv2.imread('./image/unknown_Z.png')
cv2.imshow('detected digit', mydigit)
mydigit_gray = cv2.cvtColor(mydigit, cv2.COLOR_BGR2GRAY)
mydigit_test = mydigit_gray.reshape((-1,400)).astype(np.float32)
ret, result, neighbours, dist = knn.findNearest(mydigit_test, k=5)
print(ret)

font = cv2.FONT_HERSHEY_SIMPLEX
cv2.putText(mydigit, str(int(ret)), (0, 7), font, 0.3, (255, 0, 0), 1, cv2.LINE_AA)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

2.0

In []:



Step4/4: การทดสอบแบบจำลอง - ตรวจสอบด้วยด้วยการเขียนเอง-H 0

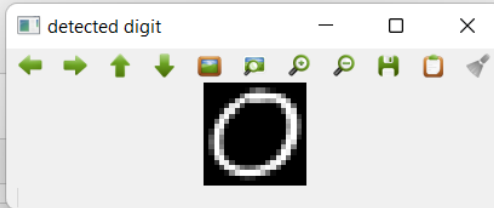
```
In [*]: mydigit = cv2.imread('./image/unknown_H.png')
cv2.imshow('detected digit', mydigit)
mydigit_gray = cv2.cvtColor(mydigit, cv2.COLOR_BGR2GRAY)
mydigit_test = mydigit_gray.reshape (-1,400).astype(np. float32)
ret, result, neighbours, dist = knn. findNearest (mydigit_test, k=5)
print(ret)

font = cv2.FONT_HERSHEY_SIMPLEX
cv2.putText(mydigit, str(int(ret)), (0, 7), font, 0.3, (255, 0, 0), 1, cv2.LINE_AA)

cv2.waitKey (0)
cv2.destroyAllWindows ()

0.0
```

In []:



Step4/4: การทดสอบแบบจำลอง - ตรวจสอบด้วยด้วยการเขียนเอง-H 1

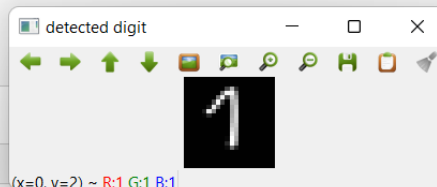
```
In [*]: mydigit = cv2.imread('./image/unknown_H.png')
cv2.imshow('detected digit', mydigit)
mydigit_gray = cv2.cvtColor(mydigit, cv2.COLOR_BGR2GRAY)
mydigit_test = mydigit_gray.reshape (-1,400).astype(np. float32)
ret, result, neighbours, dist = knn. findNearest (mydigit_test, k=5)
print(ret)

font = cv2.FONT_HERSHEY_SIMPLEX
cv2.putText(mydigit, str(int(ret)), (0, 7), font, 0.3, (255, 0, 0), 1, cv2.LINE_AA)

cv2.waitKey (0)
cv2.destroyAllWindows ()

1.0
```

In []:



Step4/4: การทดสอบแบบจำลอง - ตรวจสอบด้วยด้วยการเขียนเอง-H 2

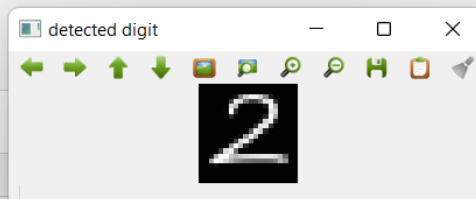
```
In [*]: mydigit = cv2.imread('./image/unknown_H.png')
cv2.imshow('detected digit', mydigit)
mydigit_gray = cv2.cvtColor(mydigit, cv2.COLOR_BGR2GRAY)
mydigit_test = mydigit_gray.reshape (-1,400).astype(np. float32)
ret, result, neighbours, dist = knn. findNearest (mydigit_test, k=5)
print(ret)

font = cv2.FONT_HERSHEY_SIMPLEX
cv2.putText(mydigit, str(int(ret)), (0, 7), font, 0.3, (255, 0, 0), 1, cv2.LINE_AA)

cv2.waitKey (0)
cv2.destroyAllWindows ()

2.0
```

In []:



Step4/4: การทดสอบแบบจำลอง - ตรวจสอบด้วยด้วยการเขียนเอง-H 3

```
In [*]: mydigit = cv2.imread('./image/unknown_H.png')
cv2.imshow('detected digit', mydigit)
mydigit_gray = cv2.cvtColor(mydigit, cv2.COLOR_BGR2GRAY)
mydigit_test = mydigit_gray.reshape (-1,400).astype(np. float32)
ret, result, neighbours, dist = knn. findNearest (mydigit_test, k=5)
print(ret)

font = cv2.FONT_HERSHEY_SIMPLEX
cv2.putText(mydigit, str(int(ret)), (0, 7), font, 0.3, (255, 0, 0), 1, cv2.LINE_AA)

cv2.waitKey (0)
cv2.destroyAllWindows ()

3.0
```

In []:



Step4/4: การทดสอบแบบจำลอง - ตรวจสอบด้วยด้วยการเขียนเอง-H 4

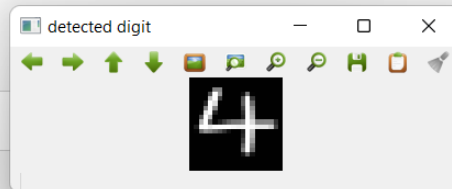
```
In [*]: mydigit = cv2.imread('./image/unknown_H.png')
cv2.imshow('detected digit', mydigit)
mydigit_gray = cv2.cvtColor(mydigit, cv2.COLOR_BGR2GRAY)
mydigit_test = mydigit_gray.reshape (-1,400).astype(np. float32)
ret, result, neighbours, dist = knn. findNearest (mydigit_test, k=5)
print(ret)

font = cv2.FONT_HERSHEY_SIMPLEX
cv2.putText(mydigit, str(int(ret)), (0, 7), font, 0.3, (255, 0, 0), 1, cv2.LINE_AA)

cv2.waitKey (0)
cv2.destroyAllWindows ()

4.0
```

In []:



Step4/4: การทดสอบแบบจำลอง - ตรวจสอบด้วยด้วยการเขียนเอง-H 5

```
In [*]: mydigit = cv2.imread('./image/unknown_H.png')
cv2.imshow('detected digit', mydigit)
mydigit_gray = cv2.cvtColor(mydigit, cv2.COLOR_BGR2GRAY)
mydigit_test = mydigit_gray.reshape (-1,400).astype(np. float32)
ret, result, neighbours, dist = knn. findNearest (mydigit_test, k=5)
print(ret)

font = cv2.FONT_HERSHEY_SIMPLEX
cv2.putText(mydigit, str(int(ret)), (0, 7), font, 0.3, (255, 0, 0), 1, cv2.LINE_AA)

cv2.waitKey (0)
cv2.destroyAllWindows ()

5.0
```

In []:



Mission 4/4:

Test Picture1 – เปลี่ยนรูปต่อไปนี้ให้เป็นรูปที่มีรหัสและชื่อของเราเอง

Test Picture2 – ใช้รูปจาก Google-4/4

Test Picture1 – เปลี่ยนรูปต่อไปนี้ให้เป็นรูปที่มีรหัสและชื่อของเราเอง



< Capture Code >

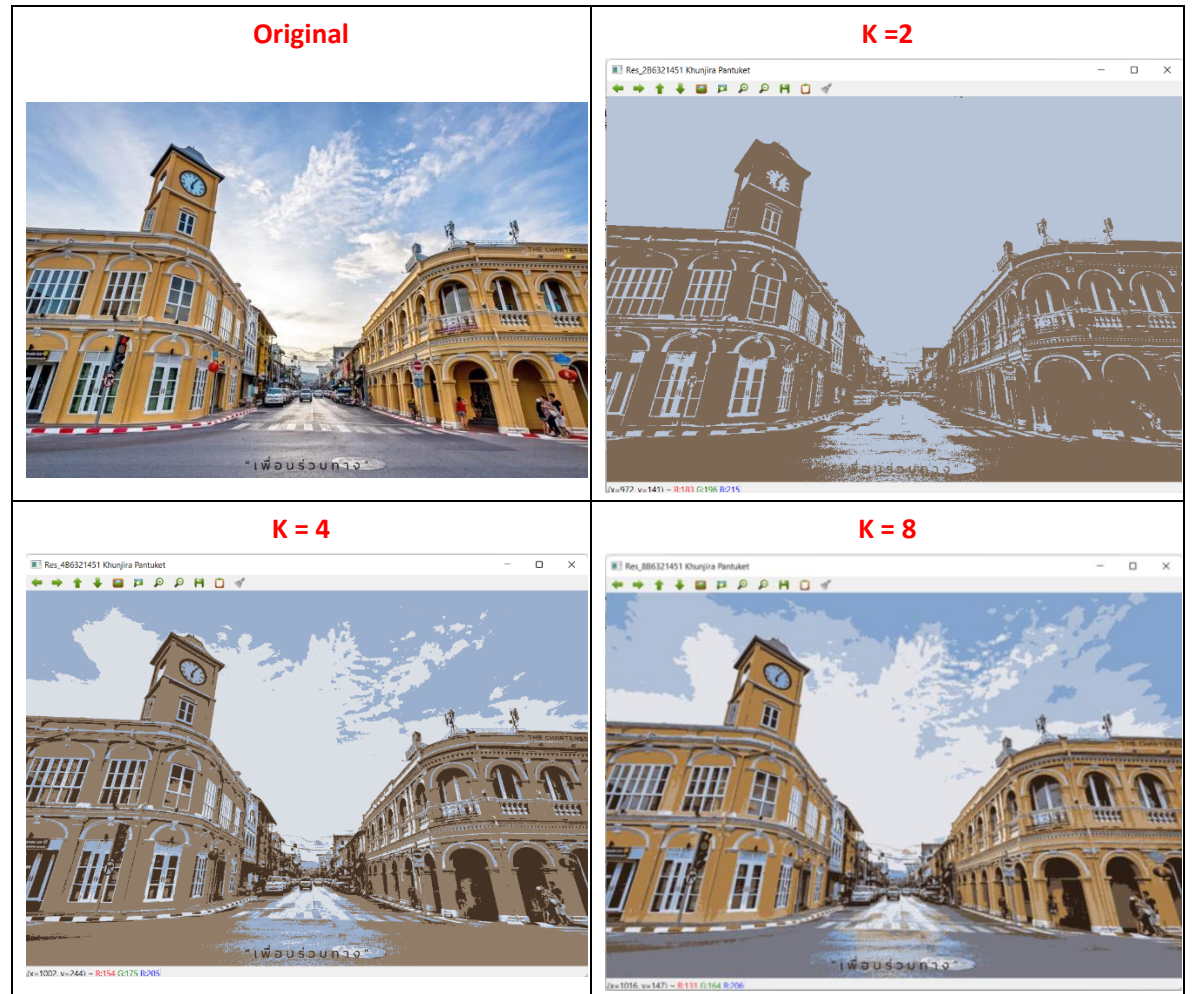
```
In [ ]: import numpy as np
import cv2
myName = "B6321451 Khunjira Pantuket"
#img = cv2.imread('./image/house.jpg')
img = cv2.imread('./image/house.jpg')
Z = img.reshape((-1,3))
Z = np.float32 (Z)
K = 8

criteria = (cv2. TERM_CRITERIA_EPS + cv2. TERM_CRITERIA_MAX_ITER, 10, 1.0)
ret, label, center=cv2.kmeans (Z,K, None, criteria, 10,cv2.KMEANS_RANDOM_CENTERS)

center = np.uint8 (center)
res = center[label.flatten ()]
res2 = res.reshape((img.shape))

cv2.imshow('Res_'+str(K)+myName, res2)
cv2.waitKey(0)
cv2.destroyAllWindows ( )
```

Test Picture2 – ใ้รูปจาก Google-4/4



< Capture Code >

```
In [ ]: import numpy as np
import cv2
myName = "B6321451 Khunjira Pantuket"
img = cv2.imread("D:\\Machine\\Week5\\me\\house_me2.jpg")
Z = img.reshape((-1,3))
Z = np.float32 (Z)
K = 8

criteria = (cv2. TERM_CRITERIA_EPS + cv2. TERM_CRITERIA_MAX_ITER, 10, 1.0)
ret, label, center=cv2.kmeans (Z,K, None, criteria, 10, cv2.KMEANS_RANDOM_CENTERS)

center = np.uint8 (center)
res = center[label.flatten ()]
res2 = res.reshape((img.shape))

cv2.imshow('Res_'+str(K)+myName, res2)
cv2.waitKey(0)
cv2.destroyAllWindows ( )
```