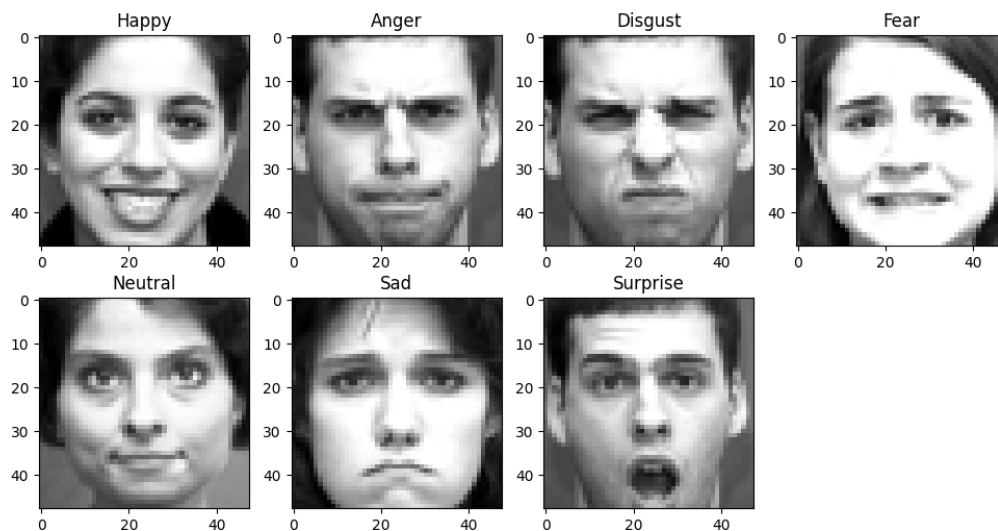


```
import matplotlib.image as mpimg

img1 = mpimg.imread('happyCK.png')
img2 = mpimg.imread('angerCK.png')
img3 = mpimg.imread('disgustCK.png')
img4 = mpimg.imread('fearCK.png')
img5 = mpimg.imread('neutralCK.png')
img6 = mpimg.imread('sadCK.png')
img7 = mpimg.imread('surpriseCK.png')
```



จะได้ เมทริกซ์ของรูปภาพ

```
[[0.25490198 0.14509805 0.14117648 ... 0.12941177 0.3254902 0.42352942]
 [0.2627451 0.13725491 0.13333334 ... 0.10980392 0.2784314 0.42352942]
 [0.23529412 0.1254902 0.12941177 ... 0.09803922 0.17254902 0.38431373]
 ...
 [0.14901961 0.01176471 0.00392157 ... 0. 0.00392157 0.03529412]
 [0.04705882 0.01568628 0. ... 0. 0.00392157 0.02745098]
 [0.03921569 0.01568628 0. ... 0. 0.00392157 0.02352941]]
```

ตัวอย่าง เมทริกซ์ รูปภาพที่ 1 (happy)

ให้ เมทริกซ์  $RA = S$

ต้องการหา เมทริกซ์  $A = R^{-1}S$

$$R = \begin{bmatrix} N & \sum_{n=1}^N n \\ \sum_{n=1}^N n & \sum_{n=1}^N n^2 \end{bmatrix}$$

$$S = \begin{bmatrix} \sum_{n=1}^N img_n \\ \sum_{n=1}^N n \times img_n \end{bmatrix}$$

$$A = R^{-1}.S$$

จะได้  $A = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$

เข้าสมการ Regression

$$y = a_1 + a_2x$$

เมื่อ  $x = \{1, 2, 3, \dots, \text{size}(img)\}$

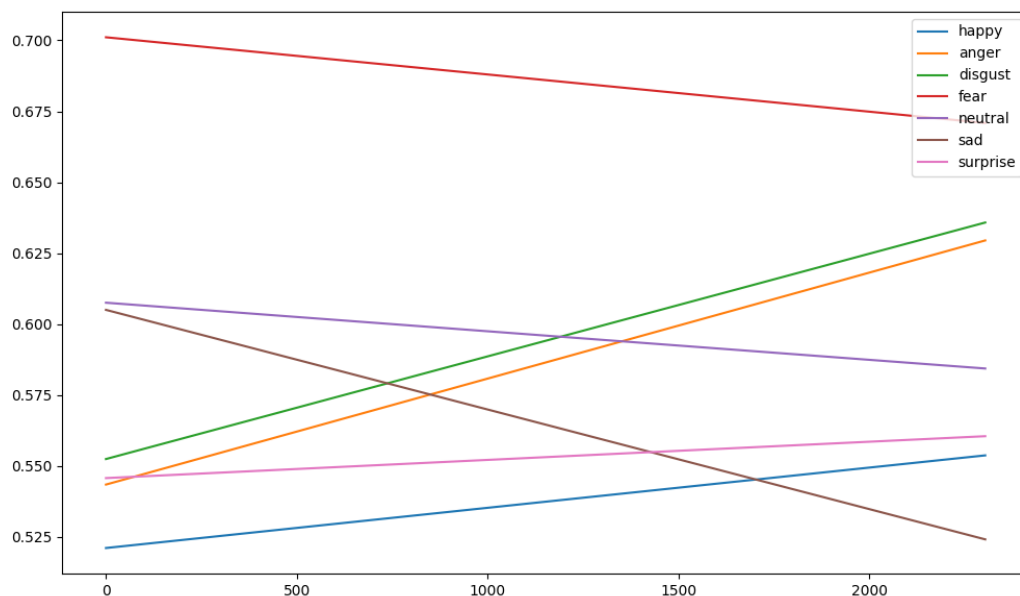
```
def create_matrix(img):
    N = len(img)
    n = np.arange(1, N+1)
    # print(len(n) == N) # True
    R = np.array([[N, np.sum(n)], [np.sum(n), np.sum(n**2)]])
    ns = np.array([n[i]*img[i] for i in range(N)])
    S = np.array([np.sum(img), [np.sum(ns)]])
    A = np.dot(inv(R), S)
    return A

n = np.arange(1, len(img1)+1)
```

```

[a1, b1] = create_matrix(img1)
y1 = a1 + b1*n
[a2, b2] = create_matrix(img2)
y2 = a2 + b2*n
[a3, b3] = create_matrix(img3)
y3 = a3 + b3*n
[a4, b4] = create_matrix(img4)
y4 = a4 + b4*n
[a5, b5] = create_matrix(img5)
y5 = a5 + b5*n
[a6, b6] = create_matrix(img6)
y6 = a6 + b6*n
[a7, b7] = create_matrix(img7)
y7 = a7 + b7*n

```



ภาพทดสอบขนาด  $48 \times 48 \text{ pixel}$

```

img_test = mpimg.imread('test.png').flatten()
[a_test, b_test] = create_matrix(img_test)
y_test = a_test + b_test*n
loss = norm(np.abs(y_test - y))

```

เมื่อเอา  $y_{test}$  ไปเทียบกับ  $y$  แล้วมี loss น้อยที่สุด ทำให้เราได้ว่ารูปภาพที่นำมา test นั้น เป็นภาพรูปหน้าอะไร เช่น happy, angry, disgusting, fear, natural, sad, surprise