

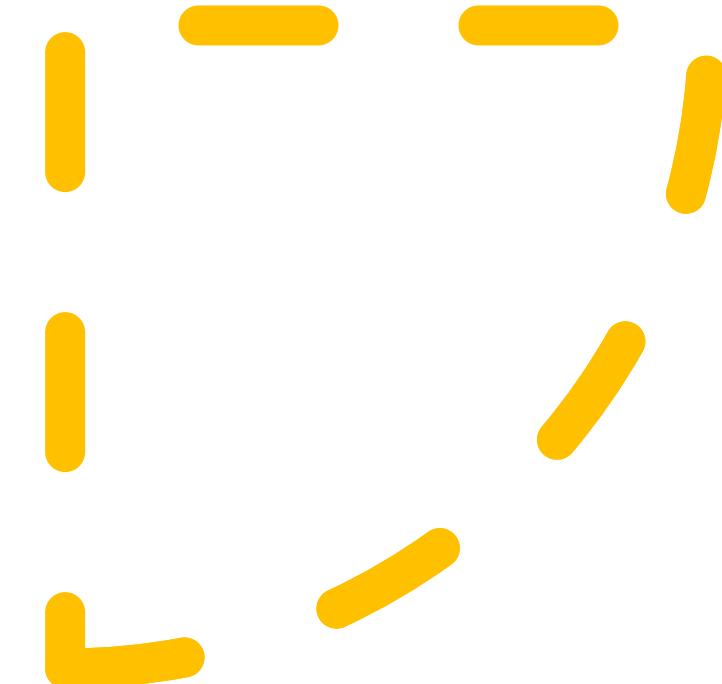
Open CV

๖๓๐๒๖๕ การประยุกต์พัฒนา



Open CV

(Open source Computer Vision) เป็นไลบรารี
ผังก์ชั้นการเขียนโปรแกรม โดยส่วนใหญ่จะมุ่งเป้าไปที่
การแสดงผลด้วยคอมพิวเตอร์แบบเรียลไทม์ (Real-
Time Computer Vision)



OpenCV

OpenCV ถูกเขียนขึ้น
ด้วยภาษา C++ มีการ
รองรับ Python, Java และ
MATLAB/OCTAVE-API



Read, Display and Write an Image

using OpenCV

Read Image

funciton สำหรับอ่านไฟล์รูปภาพใน opencv คือ cv2.imread() ใช้อ่านภาพจากไฟล์เข้ามาเป็นอาเรย์



```
1 cv.imread('image name',flag)
```



```
1 img = cv.imread('image name',IMREAD_COLOR or 1)
2 img = cv.imread('image name',IMREAD_GRAYSCALE or 0)
3 img = cv.imread('image name',IMREAD_UNCHANGED or -1)
```

Read Image

ภาพที่อ่านขึ้นมาโดยคำสั่ง `cv2.imread()` นี้หากແຕ່ໃສ່ເຊື້ອໄຟລົງໄປໂດຍໄມ່ກໍານົດວະໄຮເລຍກົງຈະອ່ານກາພມາເປັນອາເຮີຍບອນສີ 3 ສີເສມວ ມາກຕ້ອງການອ່ານກາພແບບວິນໃຫ້ກໍານົດສັ່ງກີ່ເຣີກວ່າ "ແພັກ" (flag)

Display image

ฟังก์ชันสำหรับแสดงไฟล์รูปภาพใน opencv คือ cv2.imshow() ใช้แสดงรูปภาพจากข้อมูลที่เป็นอาเรย์



```
1 cv.imshow('window_name', image)
```

Display image

cv2.imshow() ถูกออกแบบมาให้ใช้งานร่วมกับ cv2.waitKey() และ cv2.destroyAllWindows()



```
1 cv.imshow('window_name', image)
2 cv.waitKey(0)
3 cv.destroyAllWindows()
```

Write image

ដែងក្រឹងតាមរបៀបខាងក្រោមនេះ ដើម្បីរាយការណ៍ថា
ដំឡើងរូបភាពជាផាណិជ្ជកម្ម



```
1 cv.imwrite('filename', image)
```

Reading and Writing Videos

using OpenCV

Read Videos

ฟังก์ชันสำหรับอ่านวีดีโอใน opencv
คือ cv2.VideoCapture() โดยการ
อ่านไฟล์วีดีโอ จะสามารถอ่านได้ 2 แบบ
คือ

- จากไฟล์
- จาก webcam



```
1 cv.VideoCapture(path, apiPreference)
```



```
1 vid_capture = cv.VideoCapture("video/Y2Mate.mp4")
```

From a file

cv2.VideoCapture() ใช้ในการอ่านไฟล์วีดีโอ โดยจะสามารถเขียน
ได้ดังนี้

From a webcam

cv2.VideoCapture() สามารถก่อจ่องอ่านรูปภาพผ่าน webcam ได้โดยการกำหนด
path = 0



```
1 vid_capture = cv.VideoCapture(0)
```

Display videos

ในการแสดงรูปภาพโดยใช้ opencv
จะเป็นการแสดงรูปภาพที่ล่าเฟรม
โดยการเปลี่ยนอย่างรวดเร็วเพื่อให้
สามารถมองเห็นเป็นวิดีโอได้

```
1 while(vid_capture.isOpened()):  
2     ret, frame = vid_capture.read()  
3     if ret == True:  
4         cv.imshow('Frame',frame)  
5         key = cv.waitKey(33)  
6         if key == ord('q'):  
7             break  
8     else:  
9         break  
10  
11 vid_capture.release()  
12 cv.destroyAllWindows()
```

Writing Videos

เราสามารถก่อจั่งบันทึกวีดีโอจากแหล่งที่มาใดก็ได้ โดยจะต้องมีการดึงค่าต่างๆมาใช้ด้วย
ดีอ ความสูงและความกว้างของเฟรมภาพ



```
1 frame_width = vid_capture.get(cv.CAP_PROP_FRAME_WIDTH)
2 frame_height = vid_capture.get(cv.CAP_PROP_FRAME_HEIGHT)
3 frame_size = (int(frame_width), int(frame_height))
4 fps = 30
```

Writing Videos

ในการบันทึกวีดีโอนั้น จะนำค่าที่ดึงมามาใช้งานดังนี้



```
1 cv.VideoWriter(filename, apiPreference, fourcc, fps, frameSize[, isColor])
```

Fourcc



```
1 AVI: cv2.VideoWriter_fourcc('M','J','P','G')
2
3 MP4: cv2.VideoWriter_fourcc(*'XVID')
```

Writing Videos

ในการบันทึกวีดีโอจะบันทึกที่อะไร
เพรอม



```
1 while(vid_capture.isOpened()):  
2     ret, frame = vid_capture.read()  
3     if ret == True:  
4  
5         result.write(frame)  
6  
7         cv.imshow('Frame', frame)  
8         key = cv.waitKey(1)  
9         if key == ord('s'):  
10            break  
11  
12     else:  
13         break
```

Image Resizing

with OpenCV

Image Resizing

การปรับขนาดรูป ด้วย opencv จะต้องใช้คำสั่ง cv.resize() ในการปรับขนาดรูปตามความกว้างความสูงที่กำหนด



```
1 cv.resize(src, dsize[, dst[, fx[, fy[, interpolation]]]])
```

Image Resizing



```
1 cv.resize(src, dsize[, dst[, fx[, fy[, interpolation]]]])
```

src: เป็นรูปภาพอินพุตที่ต้องการ ซึ่งอาจเป็นสตริงที่มีเส้นทางของรูปภาพอินพุต (เช่น: 'test_image.png')

dsize: เป็นขนาดที่ต้องการของภาพที่ส่งออก โดยสามารถกำหนดความสูงและความกว้างใหม่ได้

fx: ตัวประกอบสเกลตามแกนนอน

interpolation: ทางเลือกในการปรับขนาดรูปภาพด้วยวิธีต่างๆ

Image Resizing



```
1 import cv2
2 import numpy as np
3
4 image = cv2.imread('image.jpg')
5 cv2.imshow('Original Image', image)
6
7 down_width = 300
8 down_height = 200
9 down_points = (down_width, down_height)
10 resized_down = cv2.resize(image, down_points, interpolation= cv2.INTER_LINEAR)
11
12 up_width = 600
13 up_height = 400
14 up_points = (up_width, up_height)
15 resized_up = cv2.resize(image, up_points, interpolation= cv2.INTER_LINEAR)
16
17 cv2.imshow('Resized Down by defining height and width', resized_down)
18 cv2.waitKey()
19 cv2.imshow('Resized Up image by defining height and width', resized_up)
20 cv2.waitKey()
21
22 cv2.destroyAllWindows()
```



สามารถ Resizing และยังรักษา
อัตราส่วนของภาพได้อย่างไร

Image Resizing



```
1 h,w,c = image.shape  
2 print("Original Height and Width:", h,"x", w)
```

Image Resizing



```
1 # Scaling Up the image 1.2 times by specifying both scaling factors
2 scale_up_x = 1.2
3 scale_up_y = 1.2
4 # Scaling Down the image 0.6 times specifying a single scale factor.
5 scale_down = 0.6
6
7 scaled_f_down = cv2.resize(image, None, fx= scale_down, fy= scale_down, interpolation= cv2.INTER_LINEAR)
8 scaled_f_up = cv2.resize(image, None, fx= scale_up_x, fy= scale_up_y, interpolation= cv2.INTER_LINEAR)
9
```

Cropping an Image

with OpenCV

Cropping an Image

ในการครอบหรือตัดรูปใน opencv จะใช้การเลือกข้อมูลมาใช้ว่าจะใช้ส่วนไหนบ้าง



```
1 cropped = img[start_row:end_row, start_col:end_col]
```

Cropping an Image



```
1 import cv2
2 import numpy as np
3
4 img = cv2.imread('test.jpg')
5 print(img.shape)
6 cv2.imshow("original", img)
7
8 cropped_image = img[80:280, 150:330]
9
10 cv2.imshow("cropped", cropped_image)
11
12 cv2.imwrite("Cropped Image.jpg", cropped_image)
13
14 cv2.waitKey(0)
15 cv2.destroyAllWindows()
```

Image Rotation and Translation

with OpenCV

Image Rotation



```
1 cv.rotate('imagename', rotateCode)
2
3 # rotateCode
4 cv.ROTATE_90_CLOCKWISE
5 cv.ROTATE_180
6 cv.ROTATE_90_COUNTERCLOCKWISE
```

Image Rotation

ในการหมุนมุมอีนๆ จะต้องใช้ การดูณเมทริกซ์ โดยถ้าให้รูปหนึ่งอยู่ที่พิกัด x y จะเขียนเมทริกซ์ได้ว่า

$$\begin{matrix} x_1 \\ y_1 \\ 1 \end{matrix}$$

$$M = \begin{matrix} a_x & b_x & c_x \\ a_y & b_y & c_y \end{matrix}$$

โดยค่า 1 คือค่า z เพราะรูปอยู่ในรูปแบบ 2 มิติ

ถ้ามีจำนวน n ตัวจะเขียนได้ในรูป

Image Rotation

เมทริกซ์ที่ใช้บิดแปลงภาพด้วย `cv2.warpAffine()` คือเมทริกซ์ขนาด 2×3

$$\begin{matrix} & x_1 & x_2 & \dots & x_n \\ x = & y_1 & y_2 & \dots & y_n \\ & 1 & 1 & \dots & 1 \end{matrix}$$

Image Rotation

เมทริกซ์ของพิกัดที่จะได้หลังแปลงนั้นจะได้มาจากการเอาเมทริกซ์การแปลงมาคูณเข้ากับเมทริกซ์พิกัดเดิม

$$\begin{aligned}\vec{X}_* &= \vec{M} \times \vec{X} \\&= \begin{bmatrix} a_x & b_x & c_x \\ a_y & b_y & c_y \end{bmatrix} \times \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ y_1 & y_2 & \cdots & x_n \\ 1 & 1 & \dots & 1 \end{bmatrix} \\&= \begin{bmatrix} a_x x_1 + b_x y_1 + c_x & a_x x_2 + b_x y_2 + c_x & \cdots & a_x x_n + b_x y_n + c_x \\ a_y x_1 + b_y y_1 + c_y & a_y x_2 + b_y y_2 + c_y & \cdots & a_y x_n + b_y y_n + c_y \\ 1 & 1 & \cdots & 1 \end{bmatrix}\end{aligned}$$

Image Rotation

ก็จะได้เมทริกซ์ของพิกัดใหม่

$$\vec{X}_* = \begin{bmatrix} x_{1*} & x_{2*} & \cdots & x_{n*} \\ y_{1*} & y_{2*} & \cdots & y_{n*} \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

Image Rotation

ค่าพิกัดใหม่ในแต่ละจุดเป็น

$$x_{i*} = a_x x_i + b_x y_i + c_x$$

$$y_{i*} = a_y x_i + b_y y_i + c_y$$

การใช้ `cv2.warpAffine()` จะเป็นการย้ายจุดทุกจุดพิกเซลบนภาพจากพิกัดเก่าไปพิกัดใหม่

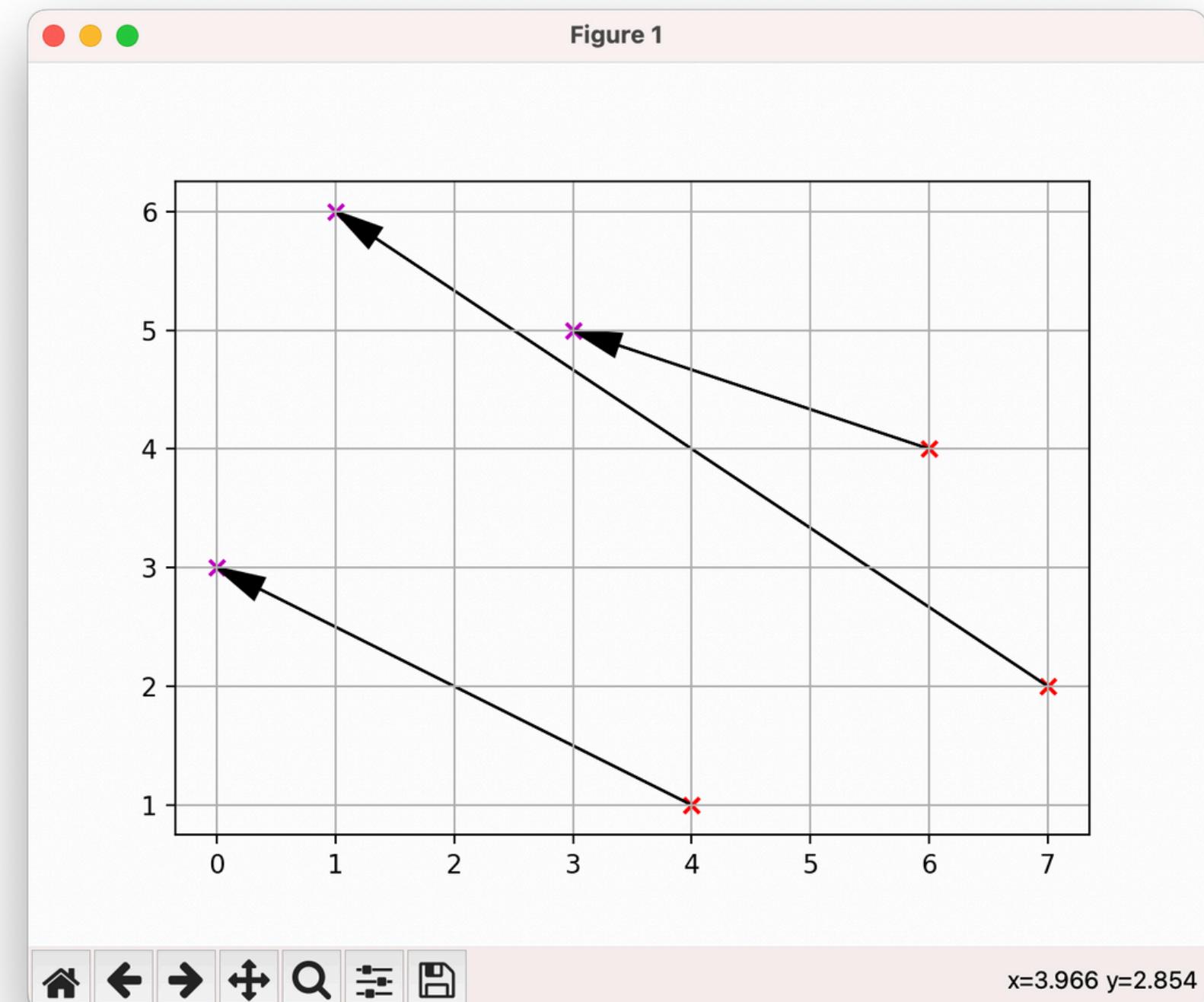


Image Rotation

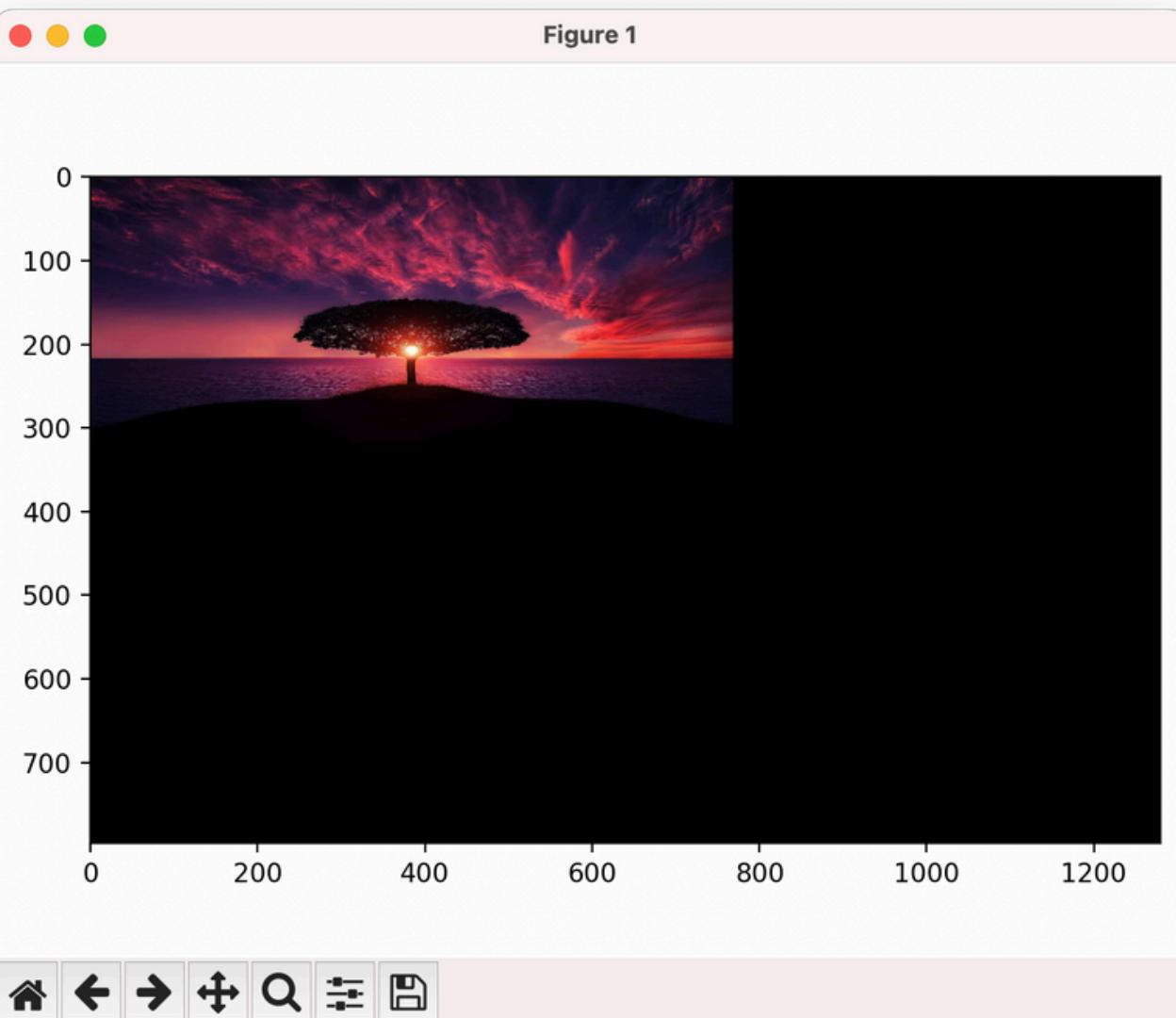


```
1 cv2.warpAffine(src, M, dsize[, dst[, flags[, borderMode[, borderValue]]]])  
2  
3 src: ภาพต้นฉบับ  
4 M: เมทริกซ์การแปลง  
5 dsize: ขนาดของภาพผลลัพธ์  
6 dst: ภาพผลลัพธ์  
7 flags: การรวมกันของวิธีการแทรกค่า เช่น INTER_LINEAR หรือ INTER_NEAREST  
8 borderMode: วิธีการประมาณค่าพิกเซลนอกขอบภาพ  
9 borderValue: ค่าที่จะใช้ในการกรณีที่ใช้ border แบบคงที่ มีค่าเริ่มต้นเป็น 0  
10
```

```
● ● ●  
1 mat = np.float32([[0.6,0,0],  
2                 [0,0.4,0]])  
3 plt.imshow(cv.warpAffine(img,mat,(1280,797))[:, :, ::-1])  
4 plt.tight_layout()  
5 plt.show()
```

$$\vec{M} = \begin{bmatrix} \text{จำนวนเท่าแนวนอน} & 0 & 0 \\ 0 & \text{จำนวนเท่าแนวตั้ง} & 0 \end{bmatrix}$$

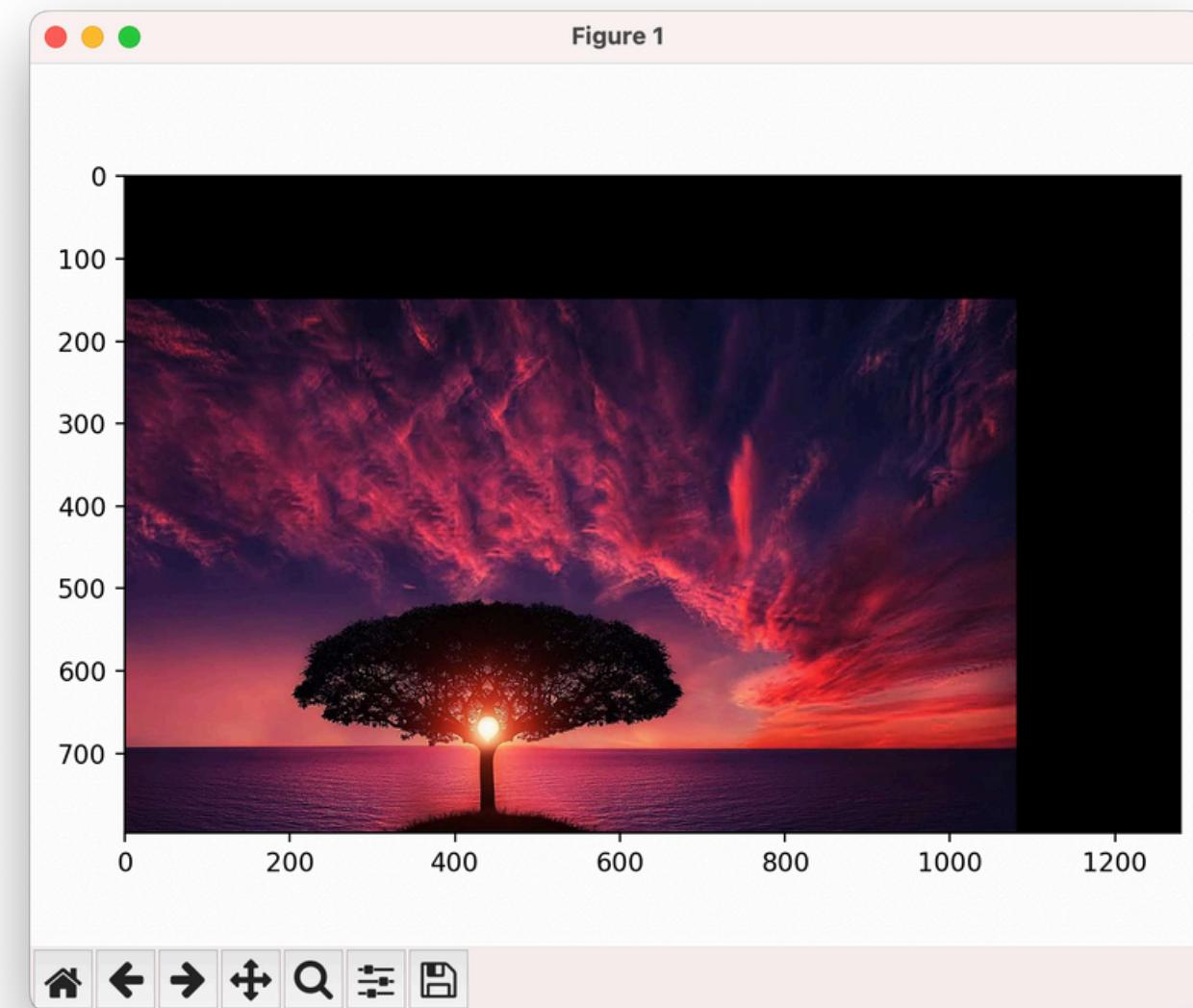
ย่อหรือขยายภาพ ใช้แค่ค่า ax กับ by



การเลื่อนตำแหน่ง

```
1 mat = np.float32([[0.6,0,0],  
2                      [0,0.4,0]])  
3 plt.imshow(cv.warpAffine(img,mat,(1280,797))[:, :, ::-1])  
4 plt.tight_layout()  
5 plt.show()
```

$$\vec{M} = \begin{bmatrix} 1 & 0 & \text{ระยะเลื่อนแนวอน} \\ 0 & 1 & \text{ระยะเลื่อนแนวตั้ง} \end{bmatrix}$$

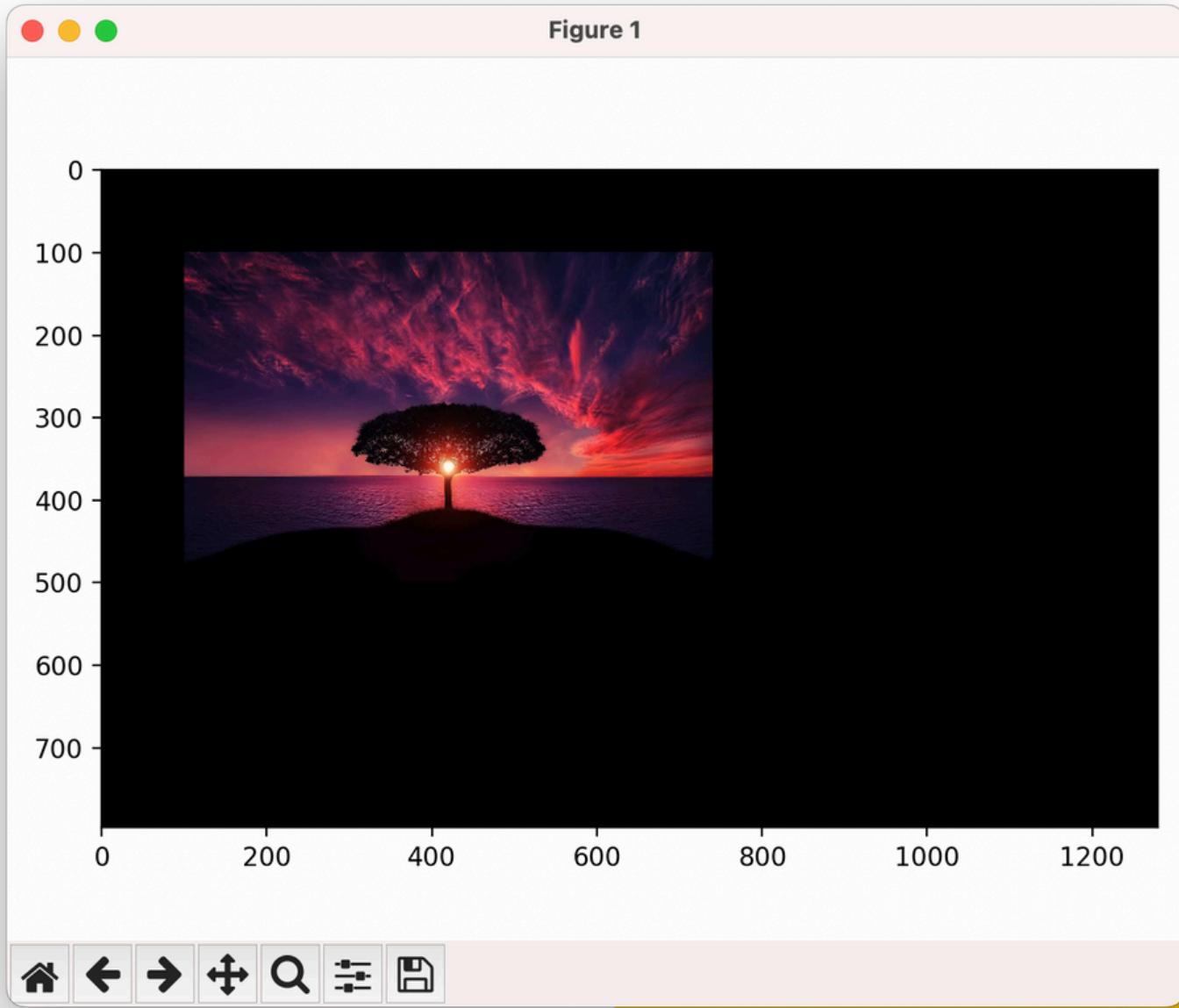




```
1 mat = np.float32([[0.5,0,100],  
2                      [0,0.5,100]])  
3 plt.imshow(cv.warpAffine(img,mat,(1280,797))[:, :, ::-1])  
4 plt.tight_layout()  
5 plt.show()
```

$$\vec{M} = \begin{bmatrix} \text{จำนวนเท่าแนวนอน} & 0 & \text{ระยะเลื่อนแนวนอน} \\ 0 & \text{จำนวนเท่าแนวตั้ง} & \text{ระยะเลื่อนแนวตั้ง} \end{bmatrix}$$

ทั้งเลื่อนตำแหน่งและย่อ^{ขยาย}



การหมุนภาพ

จะต้องใช้ เมทริกซ์

$$\vec{M} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \end{bmatrix}$$

และเมื่อไม่ได้หมุนที่จุด Center

$$\vec{M} = \begin{bmatrix} \cos\theta & -\sin\theta & c_x(1 - \cos\theta) - c_y\sin\theta \\ \sin\theta & \cos\theta & c_x\sin\theta + c_y(1 - \cos\theta) \end{bmatrix}$$

โดย θ คือขนาดของมุมที่ต้องการ
หมุนภาพ ตามเข็มนาฬิกา

การหมุนภาพ

แต่ว่าใน opencv ได้เตรียมฟังก์ชันสำหรับหมุน矩阵แบบนั้นให้โดยอัตโนมัติโดยไม่จำเป็นต้องเขียนเอง นั่นคือ cv2.getRotationMatrix2D()

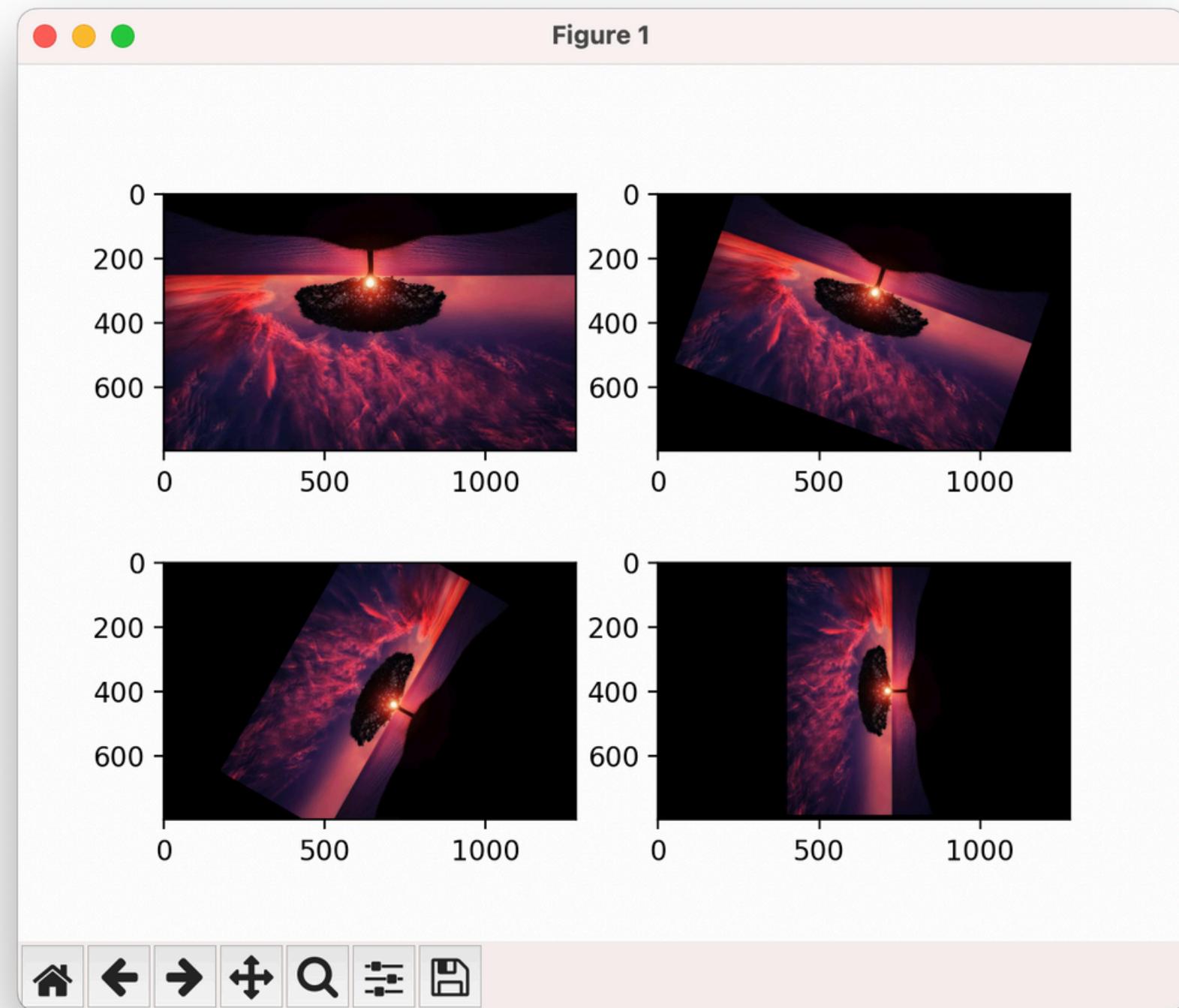


```
1 cv2.getRotationMatrix2D(center, angle, scale)
```



```
1 rotate_matrix = cv.getRotationMatrix2D(center=center, angle=45, scale=1)
2 rotated_image = cv.warpAffine(src=img, M=rotate_matrix, dsize=(width, height))
3
```

ການຄຸນາກາມ



ໂຈກຍ

- ໃຫ້ເຂີຍບໂປຣແກຣມທີ່ເລືອນກາພໄປກາງຂວາ 50 ພົກເສລ ແລະ ລົງລ່າງ 30 ພົກເສລ
- ໃຫ້ເຂີຍບໂປຣແກຣມທີ່ຮຸນກາພ 45 ອົງສາຮອບຈຸດສູນຍົກລາງຂອງ ກາພ
- ໃຫ້ເຂີຍບໂປຣແກຣມທີ່ຍັດກາພເປັນສອງເຖິງໃນແນວນອນແລະ ອົດກາພ ຄຽ້ງເໜຶ່ງໃນແນວຕັ້ງ
- ໃຫ້ເຂີຍບໂປຣແກຣມທີ່ຮວມກາຮເລືອນກາພ 100 ພົກເສລໄປກາງຂວາ, ຮຸນກາພ 30 ອົງສາ, ແລະ ຍັດກາພສອງເຖິງໃນແນວຕັ້ງ