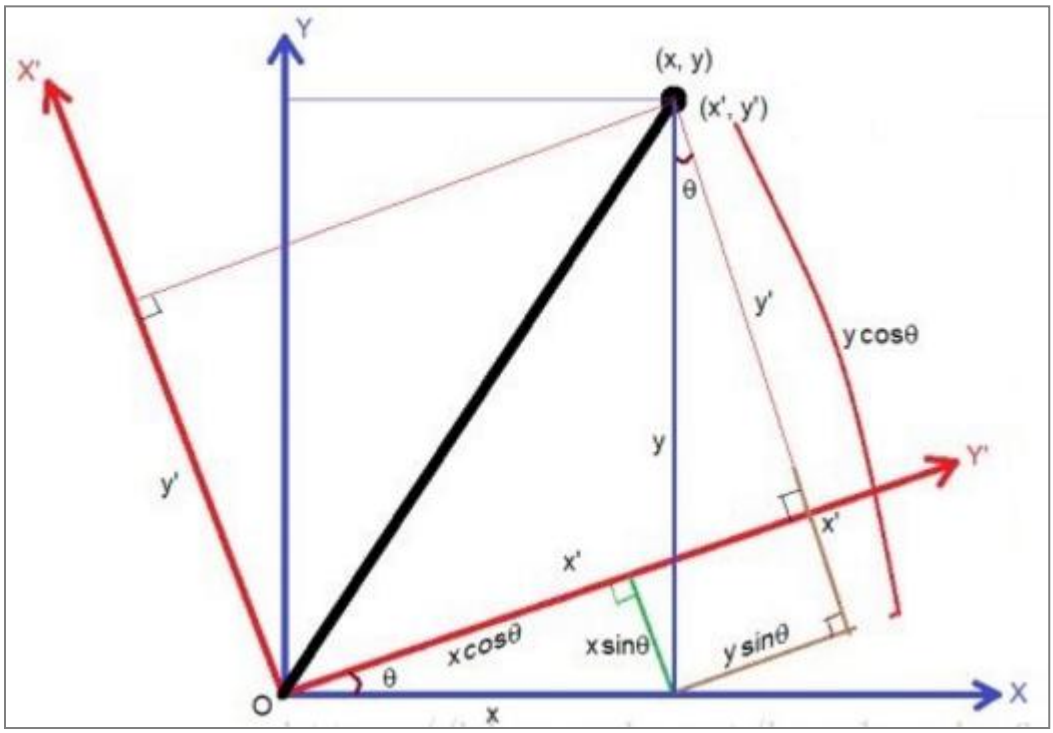
1.2.6 Picture Rotation

Image rotation is the process that an image rotates to a certain angle according to a certain position. During the rotation, the image still maintains its original size.

After the rotated image, its horizontal axis of symmetry, vertical axis of symmetry and the origin of the center coordinate may be transformed.
Therefore, we need to transform the coordinates in the image rotation accordingly.

The coordinate transformation diagram is shown below.



If the image rotates θ with counterclockwise. According to the coordinate conversion diagram, the position of the x and y axes after rotation conversion is,

$$\begin{cases} x' = r\cos(\alpha - \theta) \\ y' = r\sin(\alpha - \theta) \end{cases} \quad (1)$$

and

$$r = \sqrt{x^2 + y^2}, \sin\alpha = \frac{y}{\sqrt{x^2 + y^2}}, \cos\alpha = \frac{x}{\sqrt{x^2 + y^2}} \quad (2)$$

Take (2) into Formula (1),we can get,

$$\begin{cases} x' = x\cos\theta + y\sin\theta \\ y' = -x\sin\theta + y\cos\theta \end{cases} \quad (3)$$

After calculation,

$$[x' \quad y' \quad 1] = [x \quad y \quad 1] \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Because the gray value of the rotated picture is equal to the gray value of the corresponding position in the original picture, we can get following mathematical expression,

f(x',y')=f(x,y)

The above is the principle of rotation.

OpenCV provides an function to obtain the conversion matrix directly through the function. The function is as follows,

matRotate = cv2.getRotationMatrix2D(center, angle, scale)

- center: the center point of rotation
- angle: the angle of rotation. Positive numbers indicate counterclockwise; negative numbers indicate clockwise
- scale: transform scale (zoom size). 1 indicate no change, < 1 indicate zoom out, and > 1 indicate zoom in.

Path: /home/jetson/Dofbot\4.opencv\2.Geometric_transformation\06_Picture_Rotation.ipynb

```
import cv2
import numpy as np
img = cv2.imread(yahboom.jpg',1)
#cv2.imshow('src',img)
imgInfo = img.shape
height = imgInfo[0]
width = imgInfo[1]
matRotate = cv2.getRotationMatrix2D((height*0.5, width*0.5), 45, 1)# mat rotate 1 center 2 angle 3 scale
#100*100 25
dst = cv2.warpAffine(img, matRotate, (height,width))
```

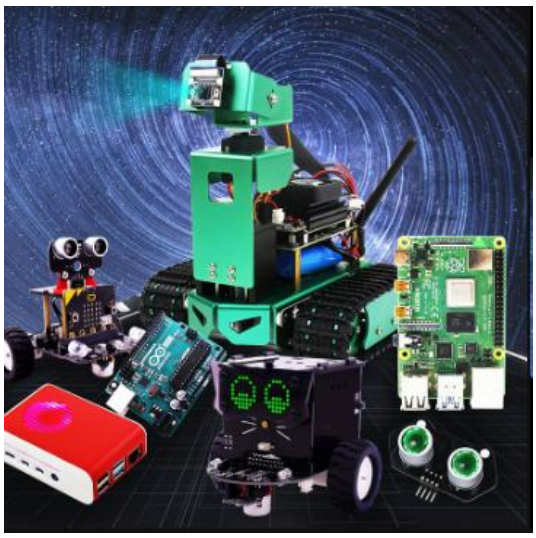
After running the following program, two images will be displayed in the jupyterLab control interface, that is original image and the rotated image.

```
#bgr8 to jpeg format
import enum
import cv2

def bgr8_to_jpeg(value, quality=75):
    return bytes(cv2.imencode('.jpg', value)[1])
```

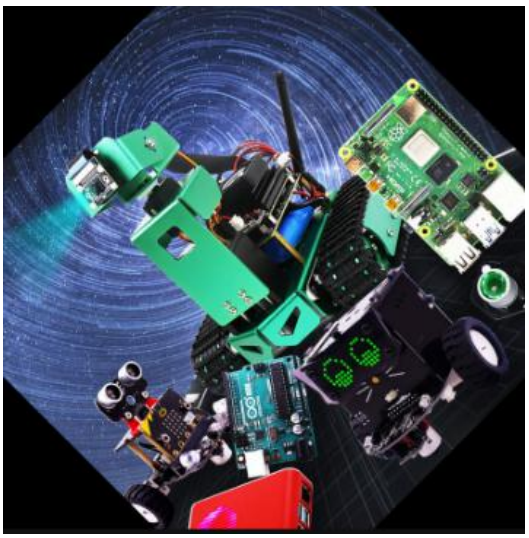
```
import ipywidgets.widgets as widgets
```

```
image_widget1 = widgets.Image(format='jpg', )
image_widget2 = widgets.Image(format='jpg', )
# create a horizontal box container to place the image widget next to eachother
image_container = widgets.HBox([image_widget1, image_widget2])

# display the container in this cell's output
display(image_container)
#display(image_widget2)

img1 = cv2.imread('image0.jpg',1)

image_widget1.value = bgr8_to_jpeg(img1)
      image_widget2.value = bgr8_to_jpeg(dst)
```

After running the following program, two images will be displayed in the jupyterLab control interface, as shown below.



[Original picture]



[Shift picture]