In [1]:

```
import numpy as np, cv2
```

Rotation Matrix

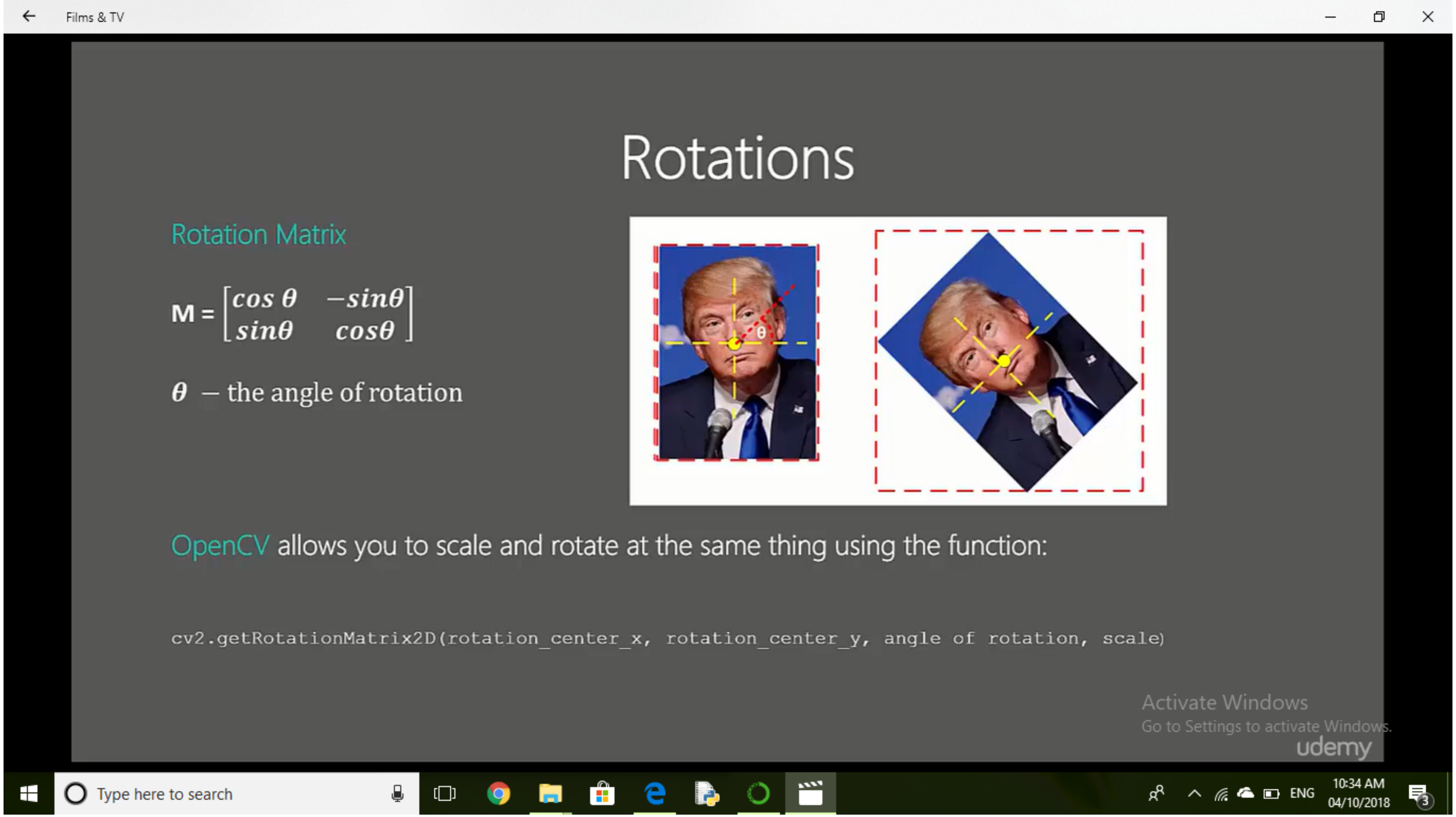
In [2]:

```
from IPython.display import Image
```

In [3]:

```
Image('tutorial images/rotation matrix.png')
```

Out[3]:



Rotations

Rotation Matrix

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

θ – the angle of rotation

OpenCV allows you to scale and rotate at the same thing using the function:

```
cv2.getRotationMatrix2D(rotation_center_x, rotation_center_y, angle of rotation, scale)
```

cv2.getRotationMatrix2D((rotation_center_x, rotation_center_y), angle of rotation, scale)

scale:- meaning ratio of actual and rotated image

Remember angle of rotation is anticlockwise

In [3]:

```python
import cv2, numpy as np
image = cv2.imread('my.JPG')
height, width = image.shape[:2]

#Divide by two to rotate the image around its center
#we assume centre of image is width /2 and height / 2
rotation_matrix = cv2.getRotationMatrix2D((width / 2, height / 2), 180, 2)
rotated_image = cv2.warpAffine(image, rotation_matrix, (width, height))
cv2.imshow('Rotated Image', rotated_image)
#cv2.imwrite('Rotated Image.png', rotated_image)
cv2.waitKey()
cv2.destroyAllWindows()
```

cv2.transpose(image)

Transpose of image

In [1]:

```python
import cv2, numpy as np
image = cv2.imread('my.JPG')
image = cv2.transpose(image)
cv2.imshow('Transposed Image', image)
cv2.waitKey()
cv2.destroyAllWindows()
```

^^it converts width -> height and height -> width