Transformation

are geometric distortions enacted upon an image. We use transformation to correct distortions or perspective issues from arising from the point of view an image was captured.

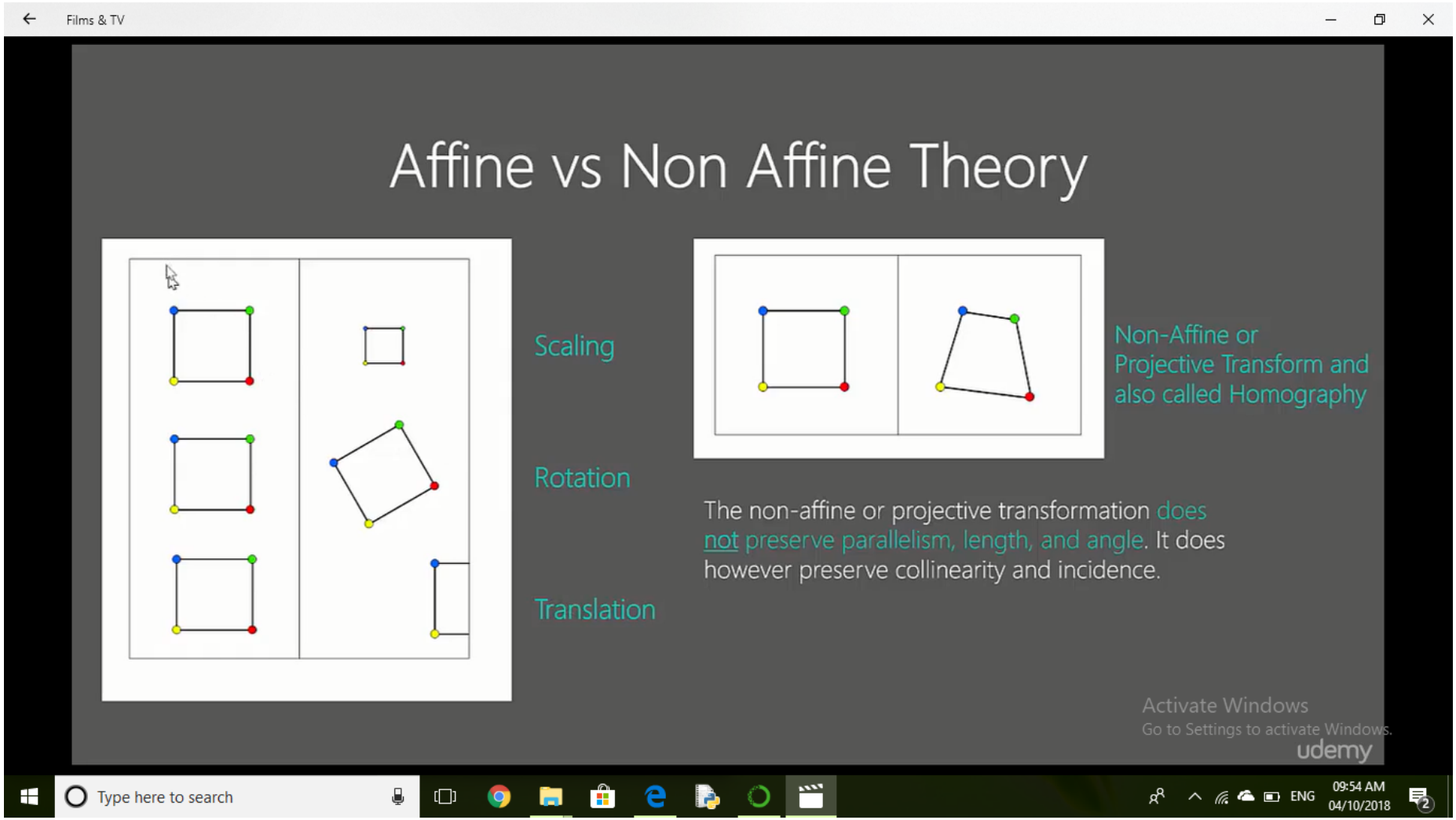
Types:

-> Affine
-> Non-Affine

In [5]:
```
1 from IPython.display import Image
```

In [6]:
```
1 Image('tutorial images/affine vs non affine.png')
```

Out[6]:



Translation

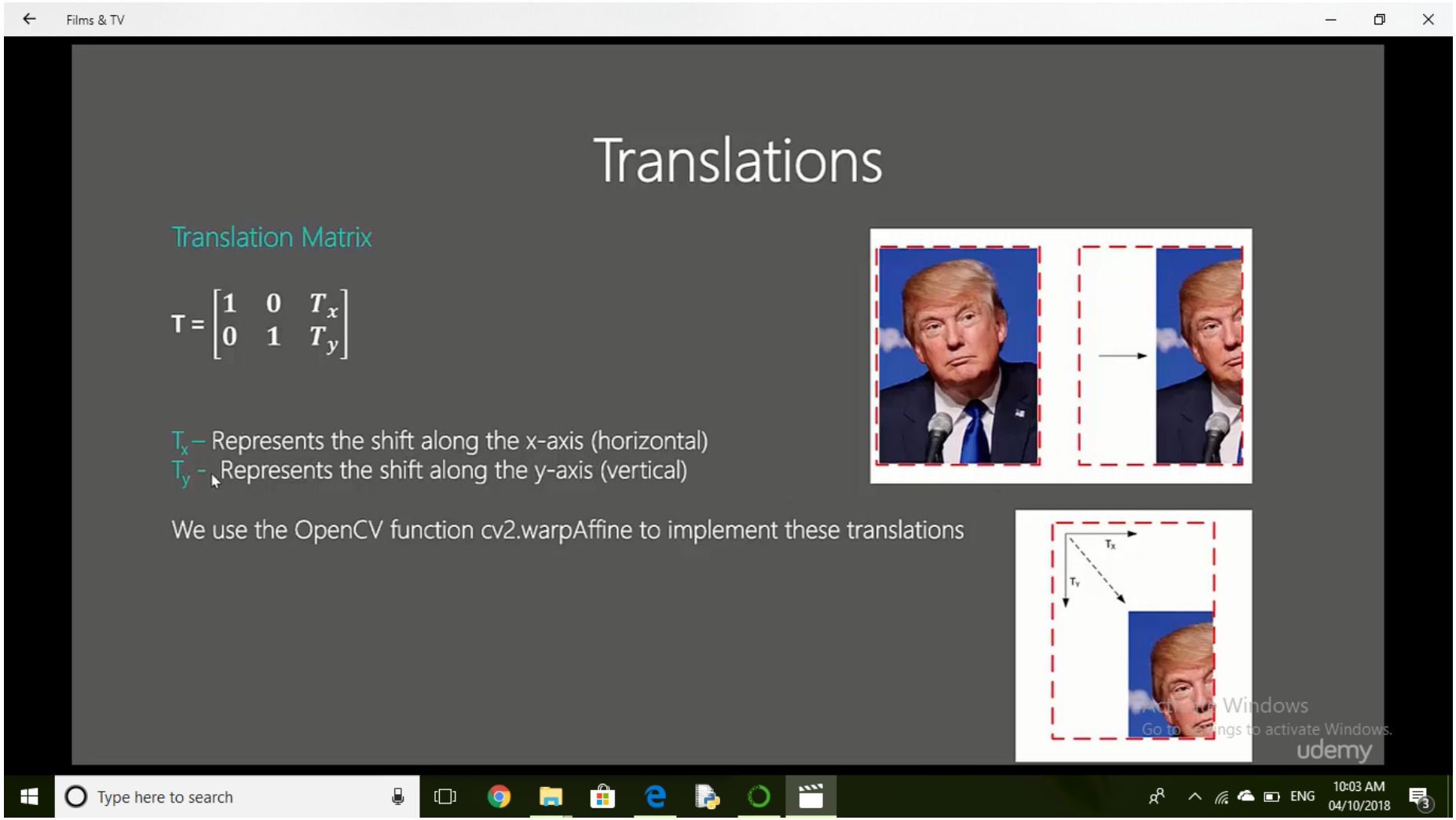
This is an affine transform that simpy shifts the position of an image. We use cv2.warpAffine to implement these transformations

warpAffine

Translation matrix

In [7]:
```
1 Image('tutorial images/translation matrix.png')
```

Out[7]:



In [18]:
```
import cv2, numpy as np
image = cv2.imread('my.JPG')
#store height and width of the image
height, width = image.shape[:2]
quarter_height, quarter_width = height / 4, width / 4

#      [1 0 Tx]
# T = [0 1 Ty]

#T is our transition matrix
T = np.float32([[1, 0, -quarter_width], [0, 1, -quarter_height]])

#We use warpAffine to transform the image using the matrix, T
img_translation = cv2.warpAffine(image, T, (width, height))
cv2.imshow('Translation', img_translation)
#cv2.imwrite('Translation.png', img_translation)
cv2.waitKey()
cv2.destroyAllWindows()
```

In [15]:
```
print(T)
```

```
[[   1.    0. -128.]
 [   0.    1. -175.]]
```

In [19]:
```
T2 = np.array([[1, 0, -333], [0, 1, 222]], np.float32)
T2
```

Out[19]:
```
array([[   1.,    0., -333.],
       [   0.,    1.,  222.]], dtype=float32)
```