In [1]:
```
1 import cv2, numpy as np
```

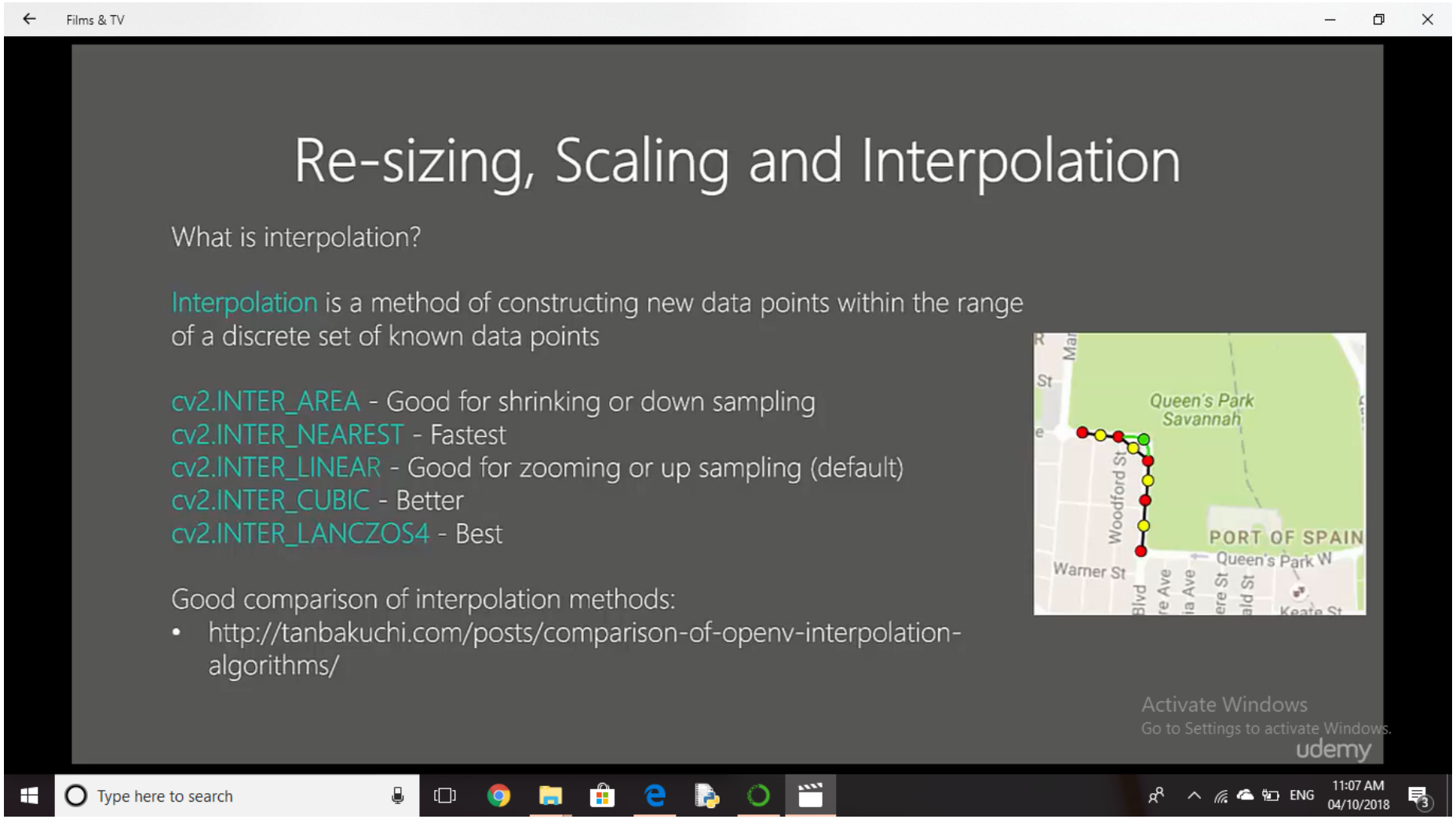
In [8]:
```
1 from IPython.display import Image
```

Scaling, re-sizing and interpolations

Re-sizing is very easy using the cv2.resize function, it's arguments are:

In [4]:
```
1 Image('tutorial images/resizing ,scaling, intepolation .png')
```

Out[4]:



cv2.resize(image, dsize(output image size), x scale, y scale, interpolate)

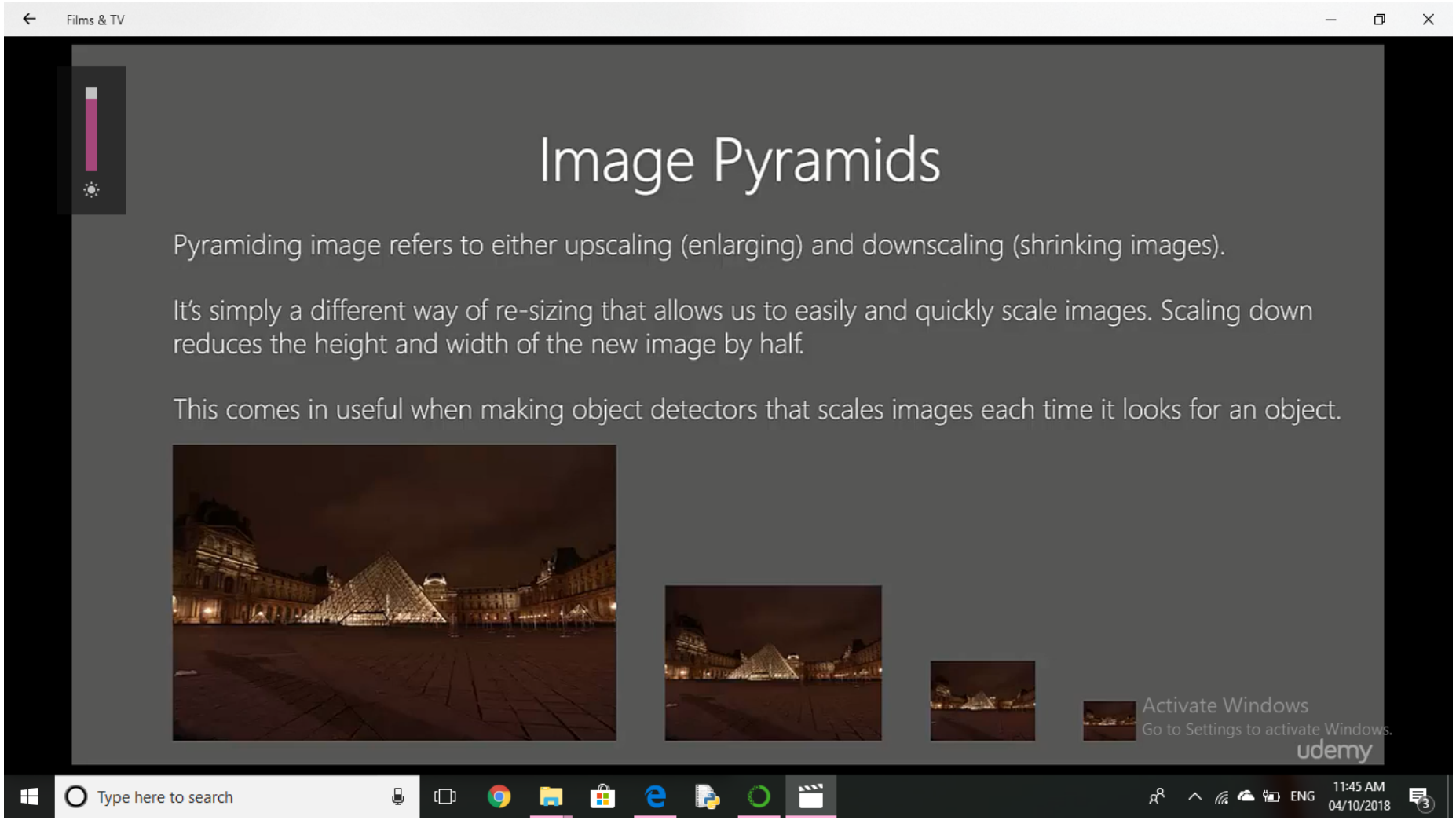
In [6]:
```
import cv2, numpy as np

#Load our input image
image = cv2.imread('my.JPG')

cv2.imshow('original image', image)
cv2.waitKey()
#Let's make our image 3 / 4 of it's original size
image_scaled = cv2.resize(image, None, fx = 0.75, fy = 0.75)

cv2.imshow('Scaling - Linear Interpolation', image_scaled)
cv2.imwrite('Scaling by Linear interpolation.png', image_scaled)
cv2.waitKey()

#Let's double the size of our image
img_scaled = cv2.resize(image, None, fx = 1.5, fy = 1.5, interpolation = cv2.INTER_CUBIC)
cv2.imshow('Scaling - Cubic Interpolation', img_scaled)
cv2.imwrite('Scaling by Cubic interpolation.png', img_scaled)
cv2.waitKey()

#Let's skew the re-sizing by setting exact dimensions

img_scaled = cv2.resize(image, (512, 512), interpolation = cv2.INTER_AREA)
cv2.imshow('Scaling Skewed size', img_scaled)
cv2.imwrite('Scaling by skewed size.png', img_scaled)
cv2.waitKey()
cv2.destroyAllWindows()
```

Image Pyramids

Useful when scaling images in object detection

In [9]:
```
1 Image('tutorial images/image pyramids.png')
```

Out[9]:



In [13]:
```
import cv2, numpy as np
image = cv2.imread('my.JPG')

cv2.imshow('Original Image', image)
cv2.waitKey()

#smaller
smaller = cv2.pyrDown(image)
cv2.imshow('Smaller image', smaller)
#cv2.imwrite('orginal to smaller.png', smaller)

#smaller ->> larger
larger = cv2.pyrUp(smaller)
cv2.imshow('small to larger image', larger)
#cv2.imwrite('original to smaller to larger.png', larger)

#original to larger
larger = cv2.pyrUp(image)
cv2.imshow('orignal to larger', larger)
#cv2.imwrite('Original to larger.png', larger)
cv2.waitKey()
cv2.destroyAllWindows()
```

^Quality of image is very low when converting original -> small -> large