**Some OpenCV Functions: -**

Here I have discussed about some functions which we can use in OpenCV for manipulating images.

In each function we will first talk a bit about it and then will see the Input code for it and its output.

# **Scaling**

Scaling is just resizing of an image. OpenCV comes with a function **cv2.resize()** for this resizing purpose.

The size of the image can be specified manually, or we can specify the scaling factor. Different interpolation methods are used: -

**cv2.INTER\_AREA** - for shrinking

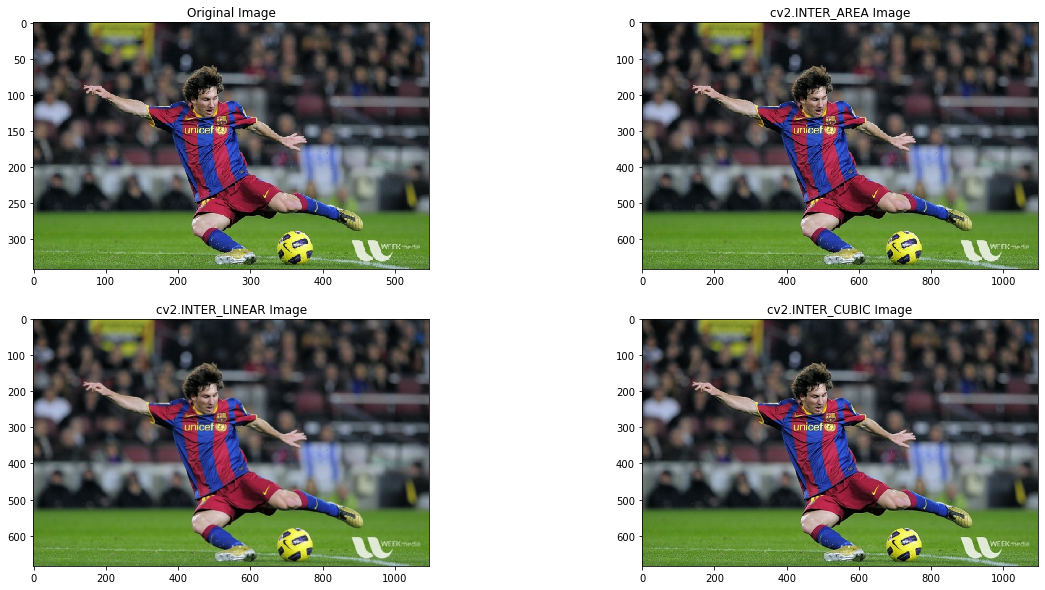
**cv2.INTER\_CUBIC** (slow) & **cv2.INTER\_LINEAR** for zooming

**By default** - interpolation method used is **cv2.INTER\_LINEAR** for all resizing purposes.

**INPUT CODE: -**

****

**OUTPUT: -**

****

So, as we can see that there isn’t much difference among all the types of Interpolations.

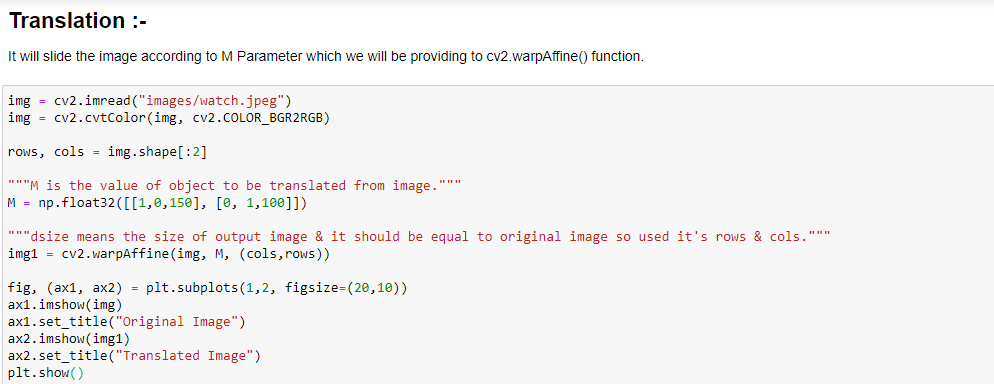
# **Translation**

Translation is the shifting of object’s location. Translation refers to the rectilinear shift of an object i.e. an image from one location to another.

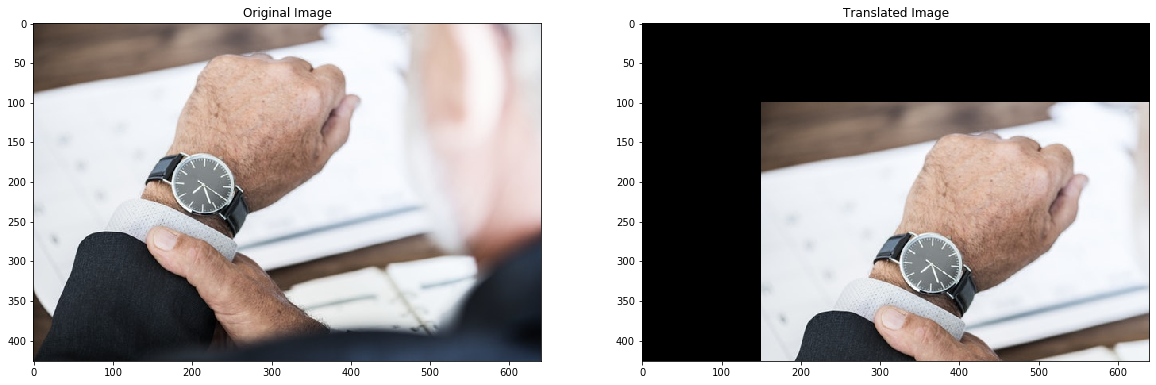
**Its advantages are: -**

* Hiding a part of the image
* Cropping an image
* Shifting an image
* Animating an image using image translations in loop.

**INPUT CODE: -**



**OUTPUT: -**



# **Rotation**

OpenCV provides a function **cv2.getRotationMatrix2D** for rotating an image in which we can even adjust the centre’s from where we want to rotate.

**INPUT CODE: -**



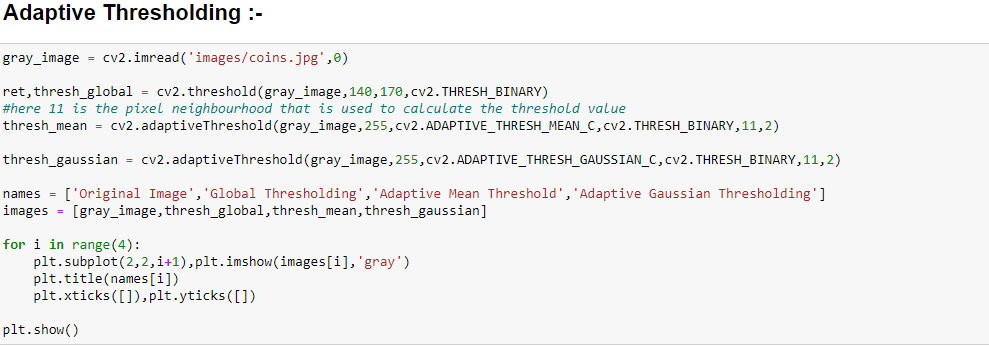
**OUTPUT: -**



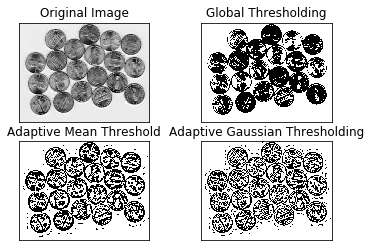
# **Adaptive Thresholding**

In case of adaptive thresholding, different threshold values are used for different parts of the image. This function gives better results for images with varying lighting conditions – hence the term “adaptive”.

**INPUT CODE: -**



**OUTPUT: -**



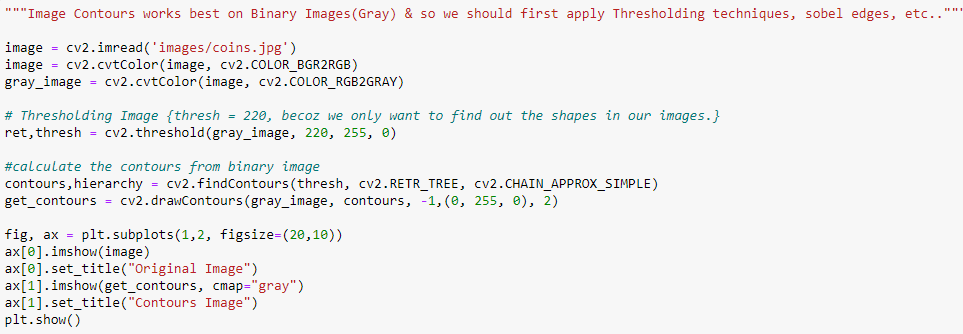
# **Image Contours**

A contour is a closed curve of points or line segments that represents the boundaries of an object in the image. Contours are essentially the shapes of objects in an image.

Unlike edges, contours are not part of an image. Instead, they are an abstract collection of points and line segments corresponding to the shapes of the object in the image.

We can use contours to count the number of objects in an image, categorize objects on the basis of their shapes, or select objects of particular shapes from the image.

**INPUT CODE: -**



**OUTPUT: -**

# 

# **Scale Invariant Feature Transform (SIFT)**

Keypoints are basically the points of interest in an image. They are locations that define what is interesting in the image. Keypoints are important, because no matter how the image is modified (rotation, shrinking, expanding, distortion), we will always find the same Keypoints for the image.

Scale Invariant Feature Transform (SIFT) is a very popular Keypoints detection algorithm. Features extracted from SIFT can be used for applications like image stitching, object detection, etc.

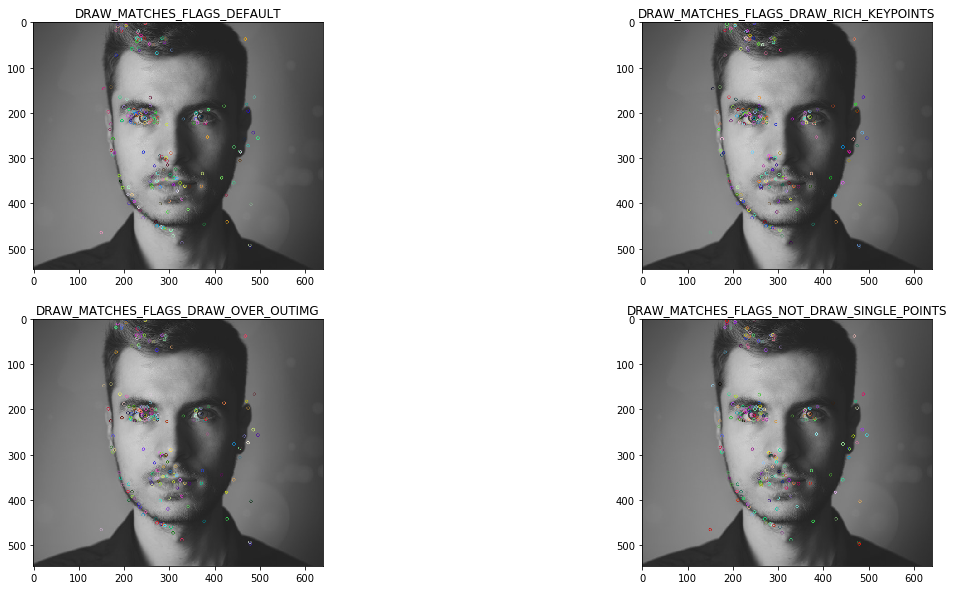
**INPUT CODE: -**



If the above code gives this error -> AttributeError: module 'cv2.cv2' has no attribute 'xfeatures2d’, then run following 2 commands: -

* !pip install opencv-python==3.4.2.17 --user
* !pip install opencv-contrib-python==3.4.2.17 –user

**OUTPUT: -**  We found 210 number of Keypoints in this image



# **Speeded-Up Robust Features (SURF)**

Speeded-Up Robust Features (SURF) is an enhanced version of SIFT. It works much faster and is more robust to image transformations. In SIFT, the scale space is approximated using Laplacian of Gaussian.

Laplacian is a kernel used for calculating the edges in an image. The Laplacian kernel works by approximating a second derivative of the image. Hence, it is very sensitive to noise.

In SURF, the Laplacian of Gaussian is calculated using a box filter (kernel). To know about box filter please refer to my **How To Blur Images With Different Filters.ipynb Notebook.**

**INPUT CODE: -**



**OUTPUT: -**



