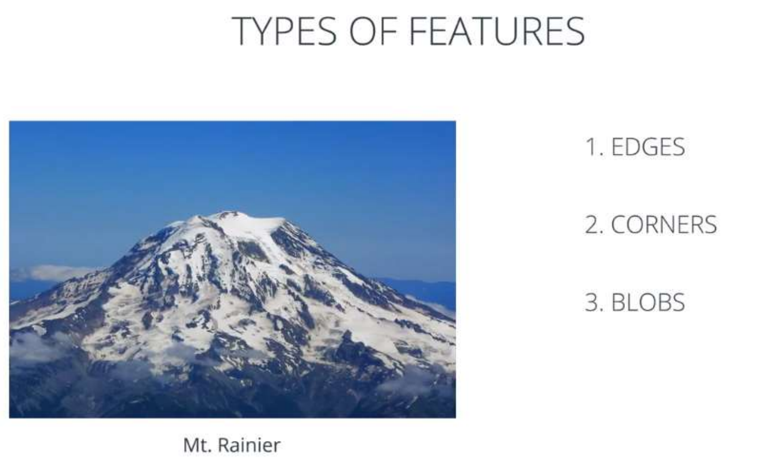
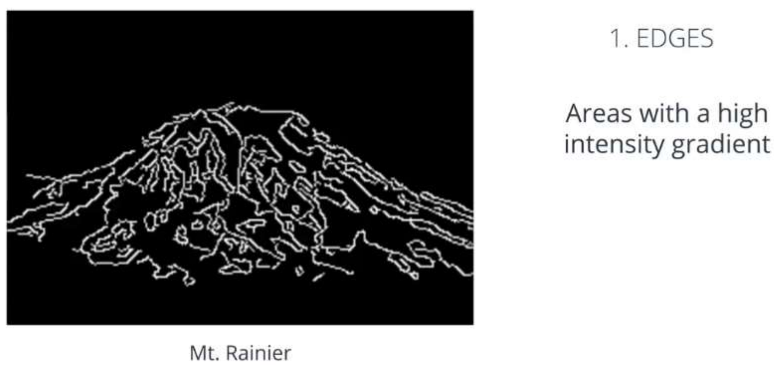
Types of Features:-

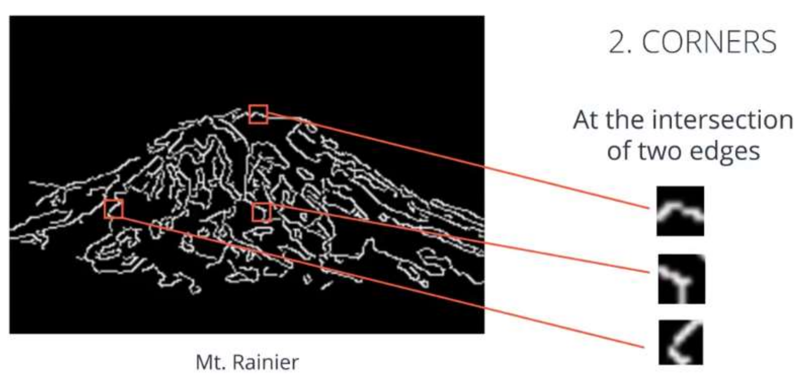
The features are only three categorical :-



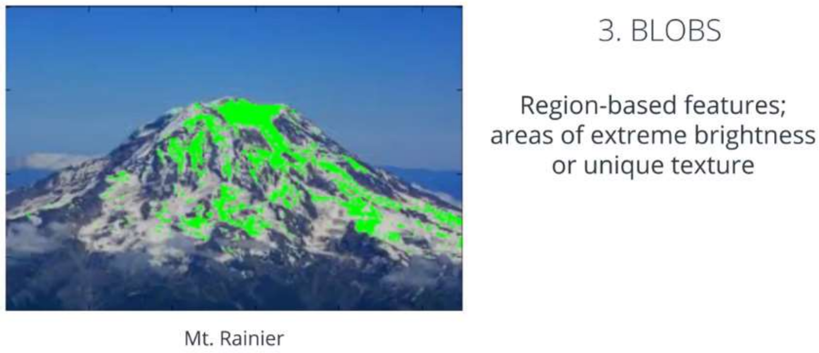
Edge :-



Corner :-



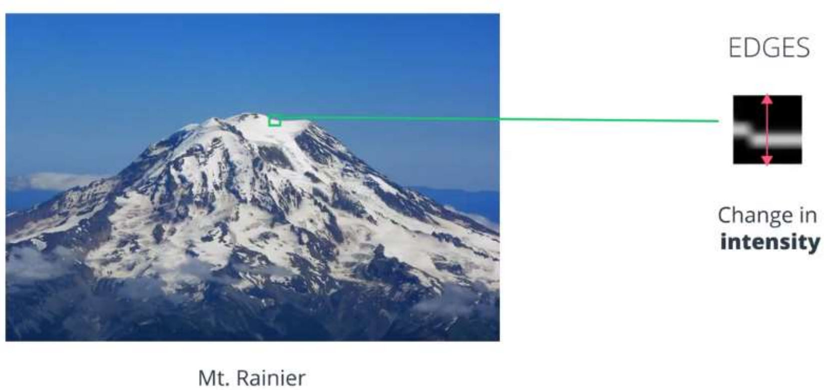
Blobs :-



Corner Detectors:-

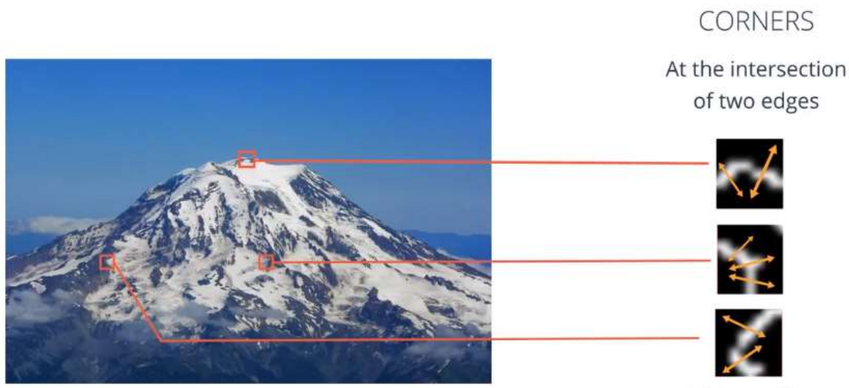
Edge :-

Region in the image with variation in intensity in one direction



Corner :-

Region in the image with large variation in intensity in all direction also we can called it ' At the intersection of two edges '



Mt. Rainier

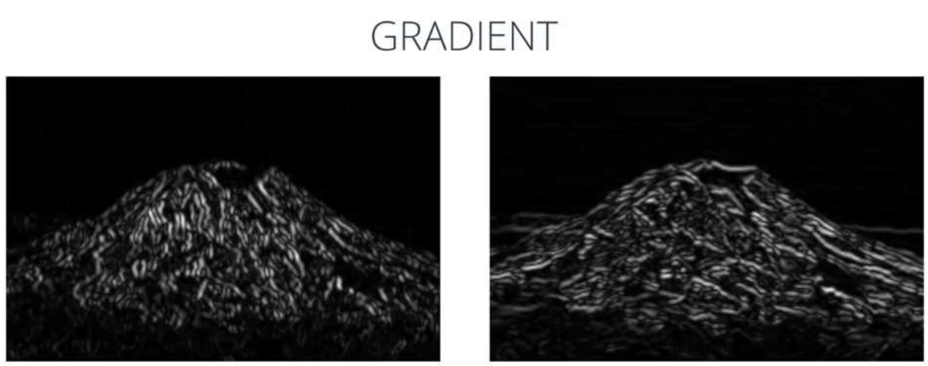
So in detect corner we need two information :-

- Magnitude
- Direction

Mt. Rainier

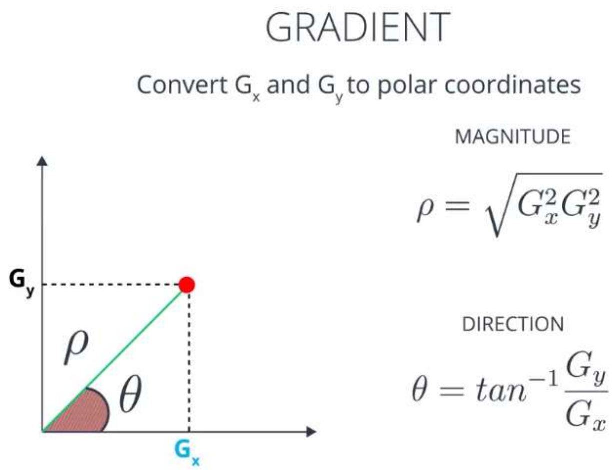
Gradient :-

To calculate the gradient we calculate it in two axis , Gx , Gy by using sobel filter

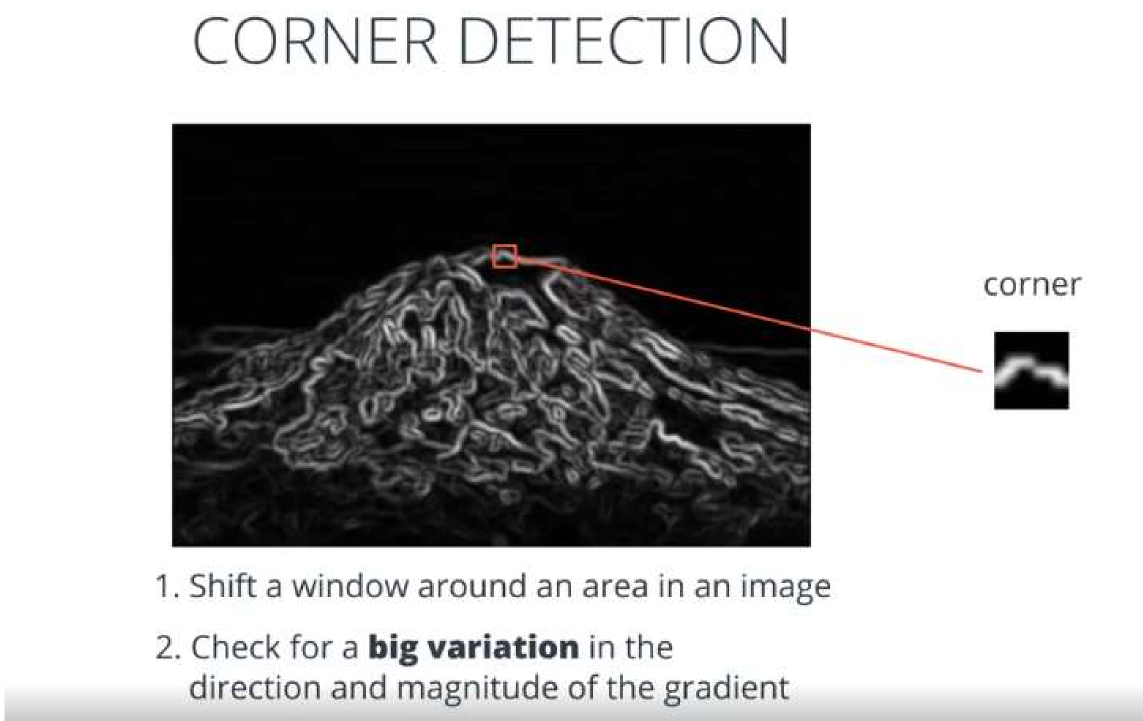


G_x G_y

Convert Gx,Gy to polar coordinate :-



Finally detect corner :-



Dilation and Erosion:-

Dilation and erosion are known as **morphological operations**. They are often performed on binary images, similar to contour detection. Dilation enlarges bright, white areas in an image by adding pixels to the perceived boundaries of objects in that image. Erosion does the opposite: it removes pixels along object boundaries and shrinks the size of objects.

Often these two operations are performed in sequence to enhance important object traits!

Dilation

To dilate an image in OpenCV, you can use the `dilate` function and three inputs: an original binary image, a kernel that determines the size of the dilation (None will result in a default size), and a number of iterations to perform the dilation (typically = 1). In the below example, we have a 5x5 kernel of ones, which move over an image, like a filter, and turn a pixel white if any of its surrounding pixels are white in a 5x5 window! We'll use a simple image of the cursive letter "j" as an example.

```
# Reads in a binary image
image = cv2.imread('j.png', 0)

# Create a 5x5 kernel of ones
kernel = np.ones((5,5),np.uint8)

# Dilate the image
dilation = cv2.dilate(image, kernel, iterations = 1)
```

Erosion

To erode an image, we do the same but with the `erode` function.

```
# Erode the image
erosion = cv2.erode(image, kernel, iterations = 1)
```

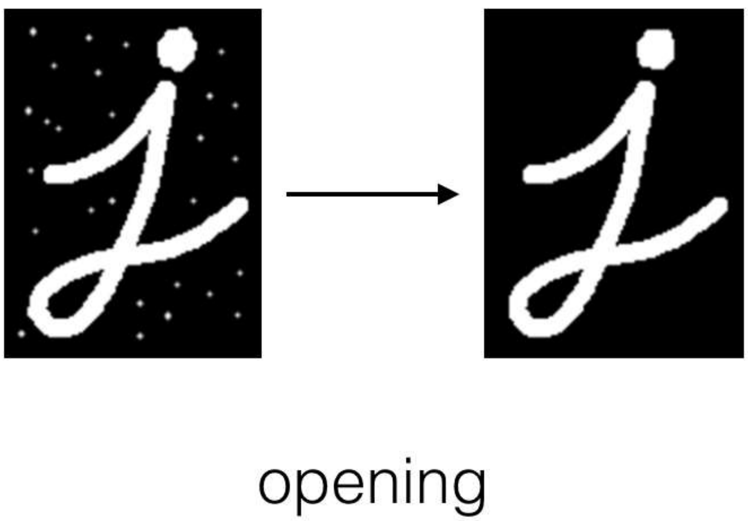
The letter "j": (left) erosion, (middle) original image, (right) dilation

Opening

As mentioned, above, these operations are often *combined* for desired results! One such combination is called **opening**, which is **erosion followed by dilation**. This is useful in noise reduction in which erosion first gets rid of noise (and shrinks the object) then dilation enlarges the object again, but the noise will have disappeared from the previous erosion!

To implement this in OpenCV, we use the function `morphologyEx` with our original image, the operation we want to perform, and our kernel passed in.

```
opening = cv2.morphologyEx(image, cv2.MORPH_OPEN, kernel)
```



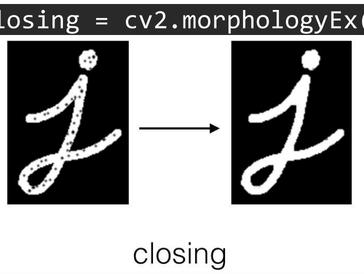
Opening

Closing

Closing is the reverse combination of opening; it's **dilation followed by erosion**, which is useful in *closing* small holes or dark areas within an object.

Closing is reverse of Opening, Dilation followed by Erosion. It is useful in closing small holes inside the foreground objects, or small black points on the object.

```
closing = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel)
```



Closing

Image Segmentation

Now that we are familiar with a few simple feature types, it may be useful to look at how we can group together different parts of an image by using these features. Grouping or segmenting images into distinct parts is known as image segmentation.

The simplest case for image segmentation is in background subtraction. In video and other applications, it is often the case that a human has to be isolated from a static or moving background, and so we have to use segmentation methods to distinguish these areas. Image segmentation is also used in a variety of complex recognition tasks, such as in classifying every pixel in an image of the road.

In the next few videos, we'll look at a couple ways to segment an image:

1. using contours to draw boundaries around different parts of an image, and
2. clustering image data by some measure of color or texture similarity.

Partially-segmented image of a road; the image separates areas that contain a pedestrian from areas in the image that contain the street or cars.

Image Contours

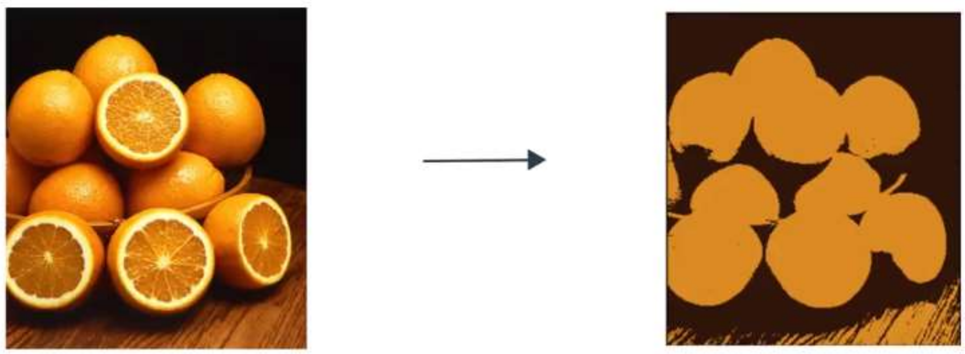
Edge detection use to detect boundary of object ‘lines and other features ’

Image Contour :-

- Continuous curves that follow the edge along a boundary
- Provide a lot of information about the shape of an object boundary
- Detected when there is a white object against a black background

K-means Clustering

Separate an image into segments by clustering data points that have similar traits.



Oranges　　　　After k-means clustering

K-means also called Unsupervised Learning :-

- A machine learning algorithm that does not rely on labeled data .
- Unsupervised learning aims to find groupings and patterns among unlabeled dataset .

K = number of clusters

k = number of clusters

k = 2　　　　k = 3　　　　k = 6

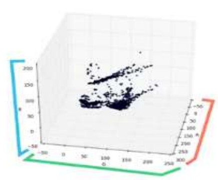
Example:-

k = 3

F(x,y) = [R, G, B]

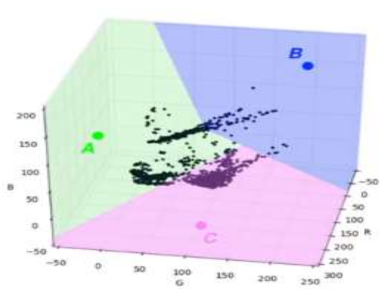
Rainbow

Every pixel in this image associated with RGB , in fact we can plot the value each pixel at the data point in RGB color space



Pixel values in RGB color space

In this example we use K = 3



1- Choose random center points (A,B,C)

2- Assign every data point to a cluster , based on its nearest center point

3- take means of all the values in each cluster

- These mean values will become the new center points