Robotics Club
IIT Kanpur

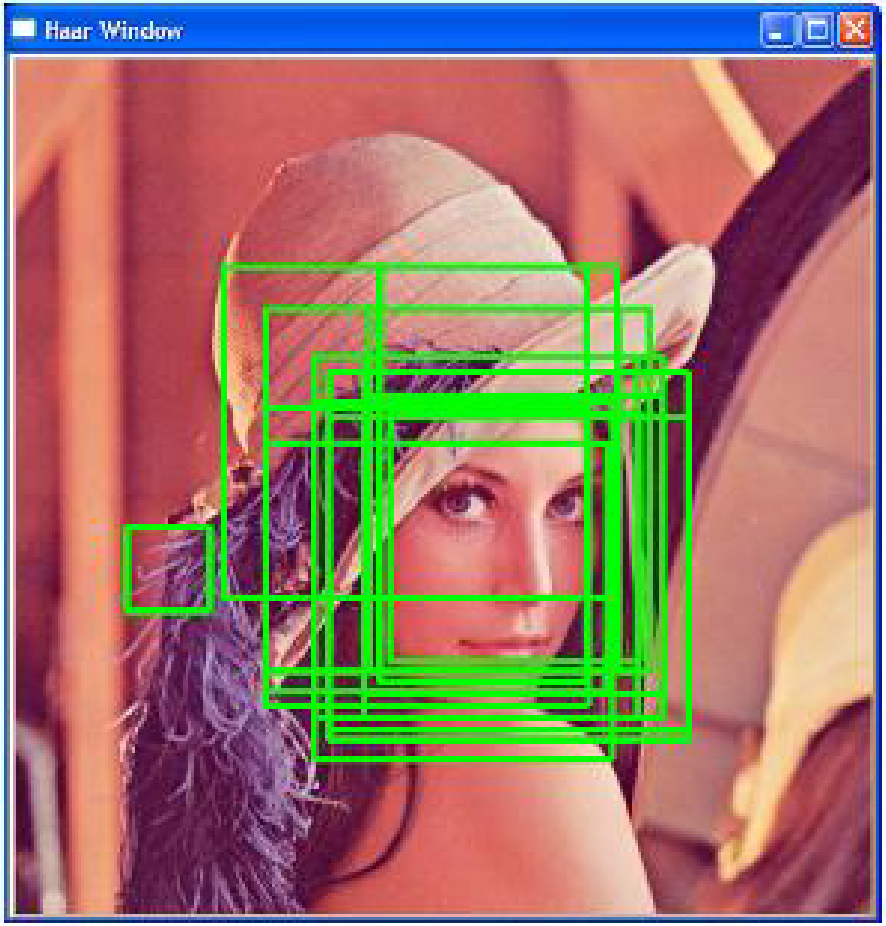
IIT KANPUR
ELECTRONICS CLUB

IMAGE PROCESSING AND OPENCV

Akshay Masare
Piyush Awasthi

WHAT IS IMAGE PROCESSING?

- IMAGE PROCESSING
  =
IMAGE + PROCESSING



WHAT IS IMAGE?

- IMAGE = Made up of PIXELS.
- Each Pixels is like an array of Numbers.
- Numbers determine colour of Pixel.

TYPES OF IMAGES :

- 1.BINARY IMAGE
- 2.GREYSCALE IMAGE
- 3.COLOURED IMAGE

BINARY IMAGE

- Each Pixel has either 1 (White) or 0 (Black)
- Depth =1 (bit)
- Number of Channels = 1

0	0	0	0	0	0	0
0	0	1	1	1	0	0
0	0	1	1	1	0	0
0	0	1	1	1	0	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

GRAYSCALE

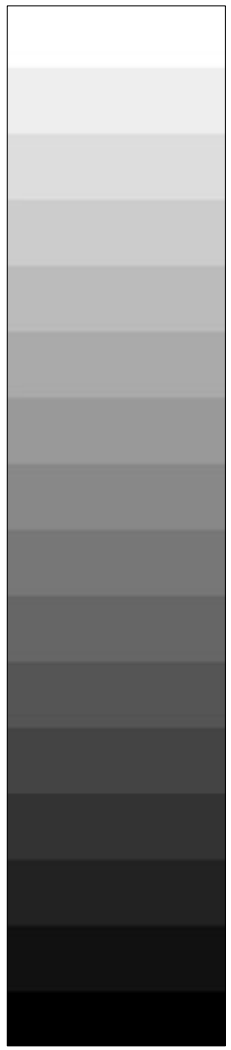
Each Pixel has a value from 0 to 255.

0 : black and 1 : White

Between 0 and 255 are shades of b&w.

Depth=8 (bits)

Number of Channels =1



GRAYSCALE IMAGE

RGB IMAGE

Each Pixel stores 3 values :-

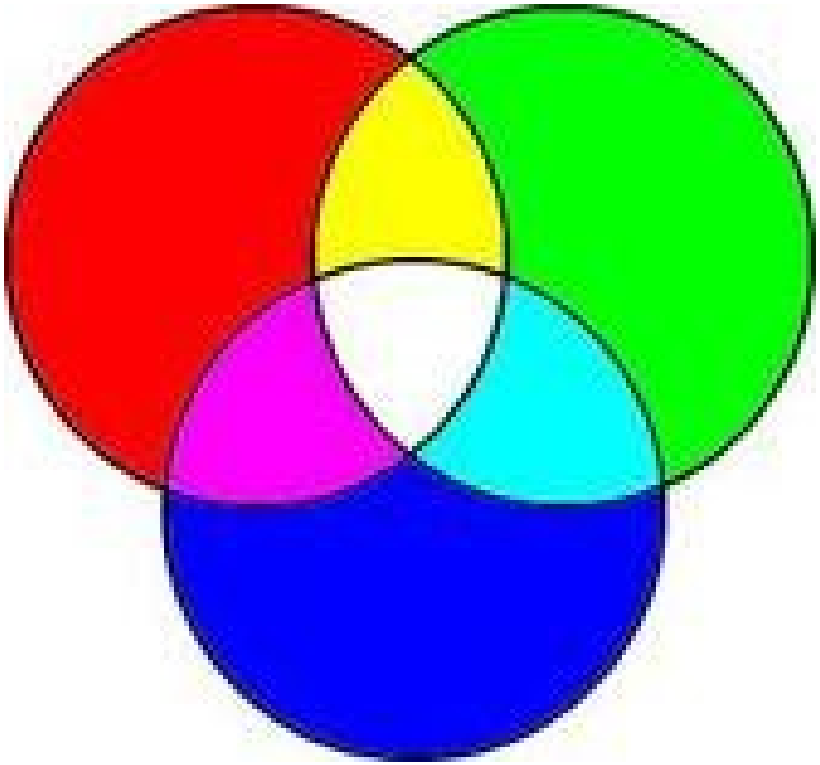
R : 0- 255

G: 0 -255

B : 0-255

Depth=8 (bits)

Number of Channels = 3



RGB IMAGE

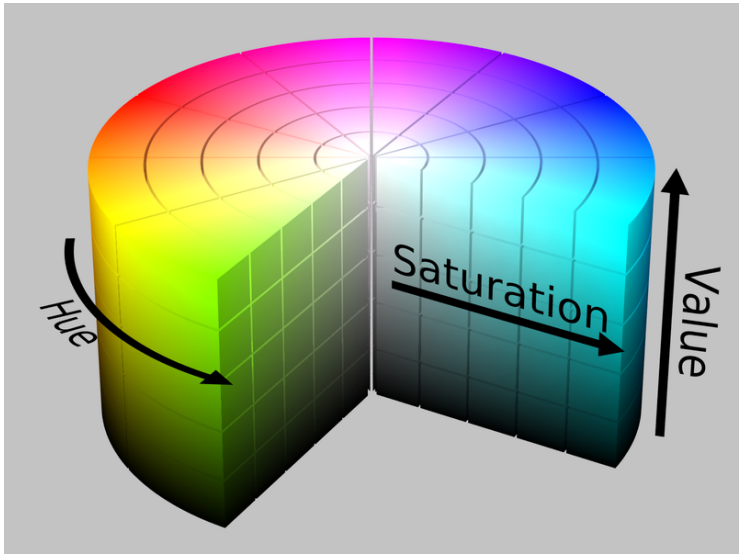
HSV IMAGE

Each pixel stores 3 values. In OpenCV

H (hue) : 0 -180

S (saturation) : 0-255

V (value) : 0-255



Depth = 8 (bits)

Number of Channels = 3

Note : Hue in general is from 0-360 , but as hue is 8 bits in OpenCV , it is shrinked to 180

OPENCV

IMAGE POINTER OR OBJECT

An image is stored as a structure *IplImage* with following elements :-

int height

int width

int nChannels

int depth

char *imageData

int widthStep

….. So on

In the newer version there is an object named *Mat* with similar elements.

imageData

An image’s data is stored as a character array whose first element is pointed by :-

Input->imageData (char pointer)

widthStep

Number of array elements in 1 row is stored in :-

input->widthStep

IMAGE POINTER

- Initialising pointer to a image (structure) :-

  IplImage input*

Mat input

- Load image to the pointer [0=gray;1=colored]

  input=cvLoadImage("apple.jpg",1)

  input=imread("apple.jpg",1)

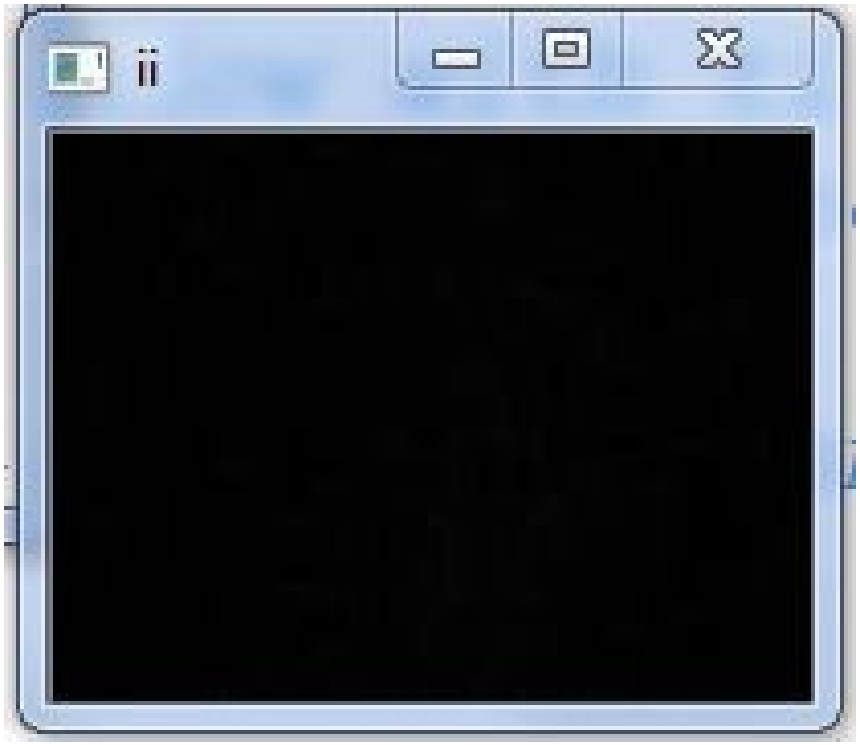
Note :The image apple.jpg must be in same folder where you save your C program

cvNamedWindow("ii",1)

namedWindow(“ii”, 1)

Creates a window named ii

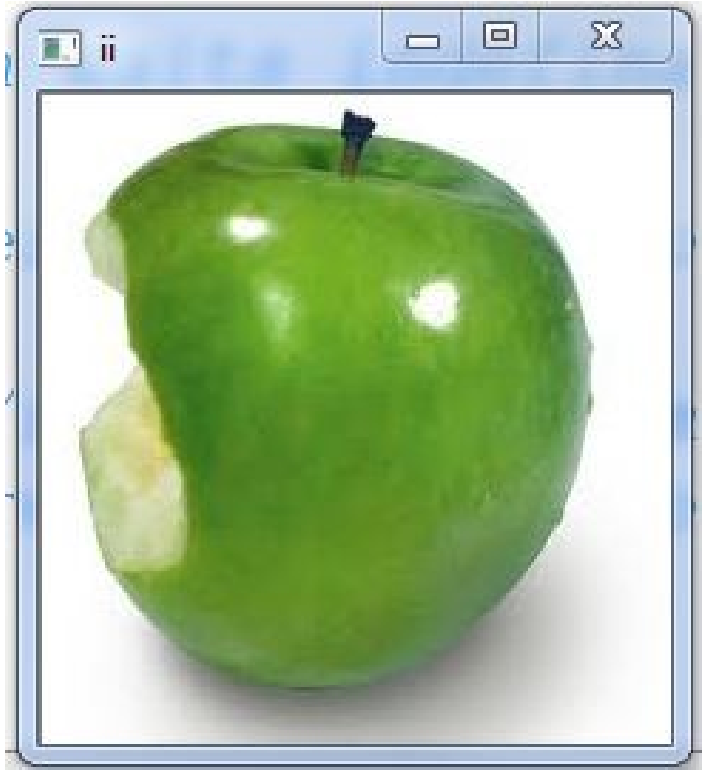
1 = Coloured 0 = Grayscale



cvShowImage("ii",input)

imshow("ii",input)

Shows image pointed by input , in the window named ii



cvWaitKey(*a number*)

If 0 or negative number is given as input:-
Waits indefinitely till key press and returns the ASCII value of the key pressed

If positive number is given as input :-
Waits for corresponding milliseconds.

CREATE AN IMAGE

To create an image you need to specify its :-

- Size (height and width)
- Depth
- Number of Channels

output=cvCreateImage(cvGetSize(input),IPL_DEPTH_8U, 3)

Mat output(2,2,CV_8UC3,Scalar(0,0,255));

TAKING IMAGE FROM CAMERA

```
CvCapture *capture=cvCreateCameraCapture(0);
for(int i=0;i<100000000;i++);
if(capture!=NULL)
                    IplImage
*frame=cvQueryFrame(capture);
```

Note : Here for loop is used to compensate time of initialization of camera in Windows

Command	Function
cvDestroyWindow("ii")	Destroys window named *ii*
cvReleaseImage(&input)	Releases image pointer *input* from memory
output=cvCloneImage(input) input.copyTo(output)	Copies image from input to output
cvSaveImage("output.jpg",output) Imwrite(“output.jpg”,output)	Saves image pointed by output naming it output

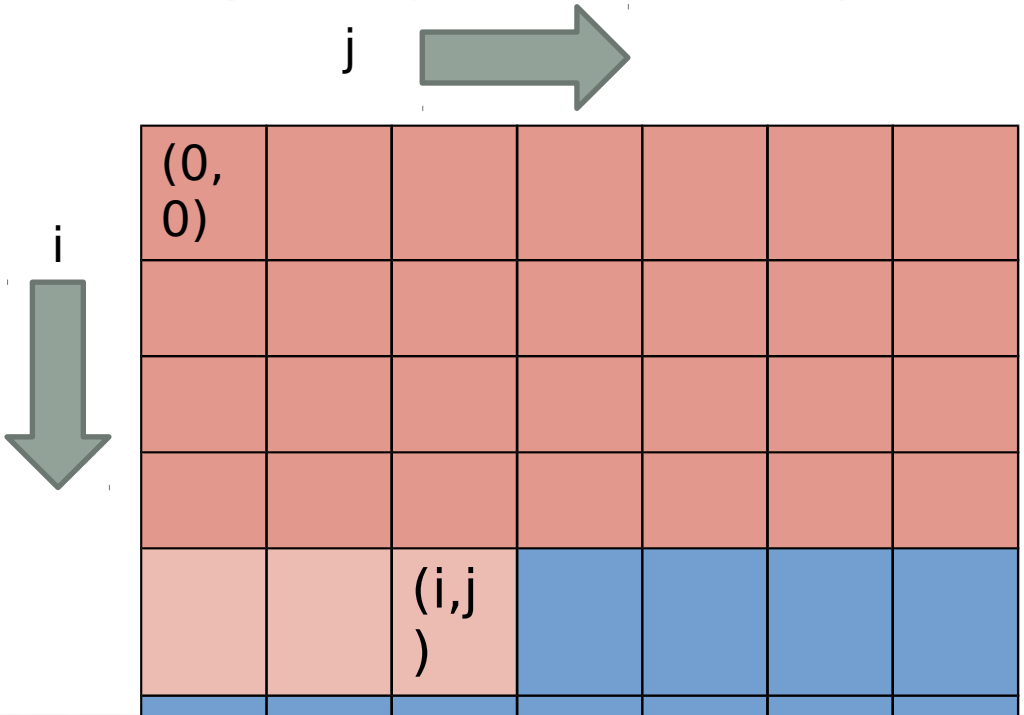
CONVERTING COLOR SPACES

- cvCvtColor(input, output, conversion type)
- cvtColor(input,output, code, 0)
- Conv. type : CV_BGR2GRAY ,CV_BGR2HSV
- Saves input image in output pointer in other color space

ACCESSING (I,J) PIXEL OF AN IMAGE

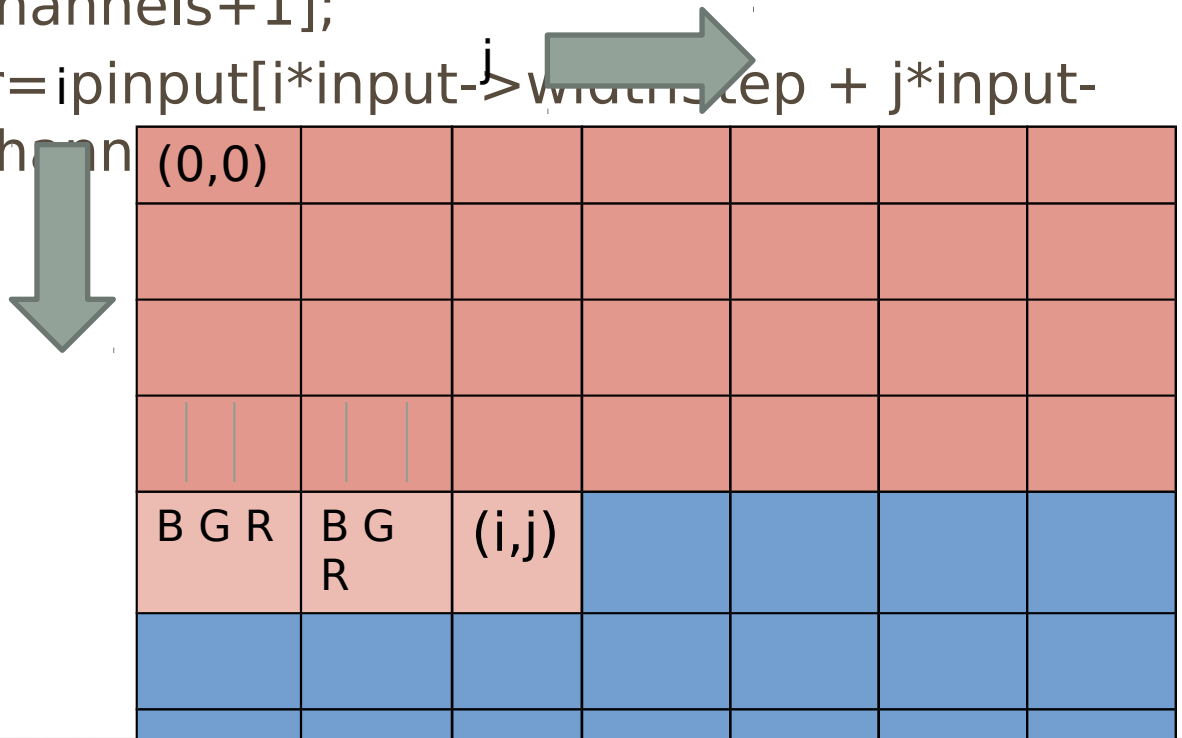
- Grayscale

```
uchar *pinput   =  (uchar*)input->imageData;
int c = pinput[i*input->widthStep + j];
Scalar intensity=img.at<uchar>(y,x);
```



3 CHANNEL IMAGE (BGR):-

```
uchar *pinput   =  (uchar*)input->imageData;
int b= pinput[i*input->widthStep + j*input->nChannels+0];
int  g= pinput[i*input->widthStep + j*input->nChannels+1];
int  r=ipinput[i*input->widthStep + j*input->nChannels+2];
```



3 CHANNEL IMAGE (BGR):-

- Vec3b intensity=img.at<Vec3b>(y,x);
- Uchar blue = intensity.val[0];
- Uchar green = intensity.val[1];
- Uchar red = intensity.val[2];

cvThreshold(input, output, threshold, maxValue, thresholdType)

Threshhold types:-

- CV_THRESH_BINARY
  max value if more than threshold, else 0
- CV_THRESH_BINARY_INV
  0 if more than threshold , else max value
- CV_THRESH_TRUNC
  threshold if more than threshold,else no change
- CV_THRESH_TOZERO
  no change if more than threshold else 0
- CV_THRESH_TOZERO_INV
  0 if more than threshold , else no change

SMOOTHING IMAGES

Image smoothing is used to reduce the the sharpness of edges and detail in an image.

- OpenCV includes most of the commonly used methods.

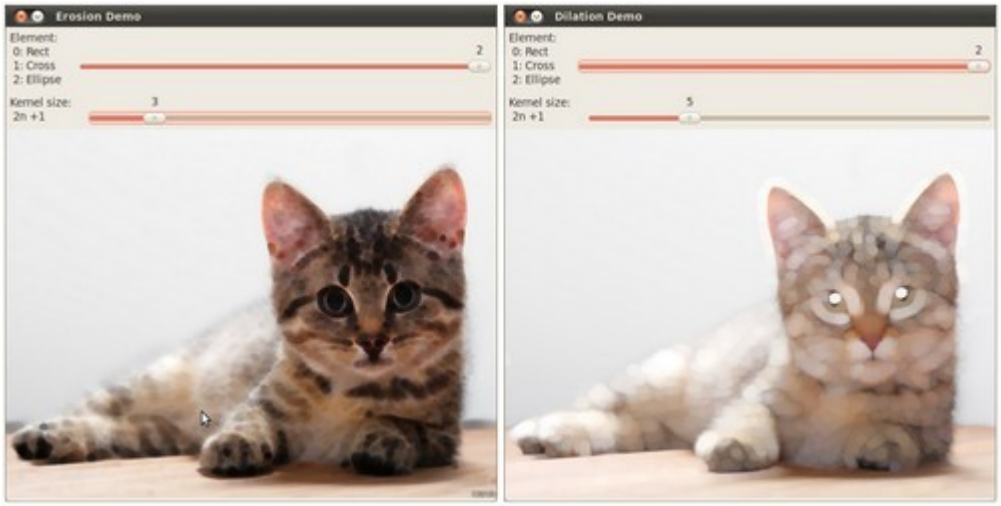
- void GaussianBlur(imagein, imageout, Size ksize,sig);

- void blur (imagein, imageout, Size ksize);

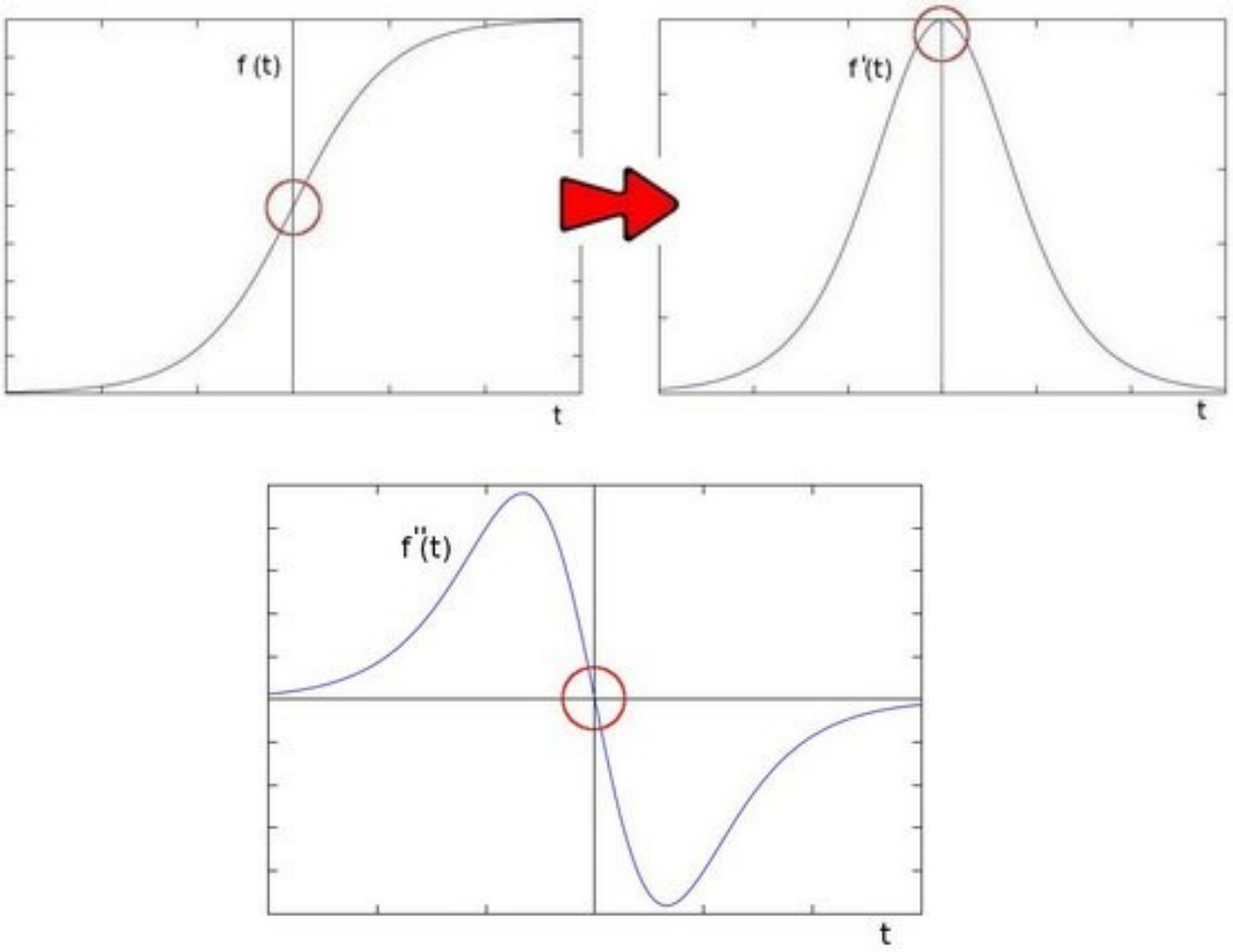
IMAGE MORPHOLOGY

- cvDilate(input , output, NULL, iterations)
- Dilate(input, output, element, anchor, iterations, border type, border value)
- Dilates an image for given number of iterations and saves it in output
- cvErode(input,erode,NULL,iterations);
- Note:here NULL is a structural element
- Erodes an image for given number of iterations and saves it in output

EFFECTS OF ERODE AND DILATE



LAPLACE OPERATOR



QUESTIONS

THANK YOU