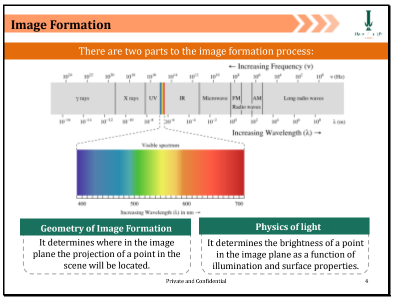
The Computer Vision Class Presentation has a good amount of Math content where the faculty is expected to break it down into simple understandable terms. This document is an attempt to facilitate the faculty with some more points on each of these slides, which will help them in explaining the concept in a better and meaningful way.

Slide 4



Electromagnetic spectrum showing the range of visible light. VIBGYOR – showing the entire range of visible light with the wavelength of the spectrum.

Slide 5

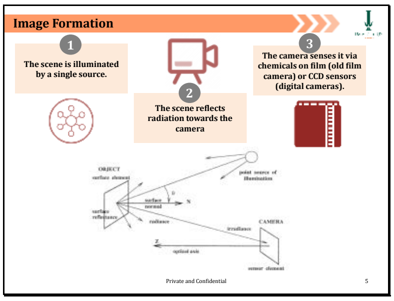
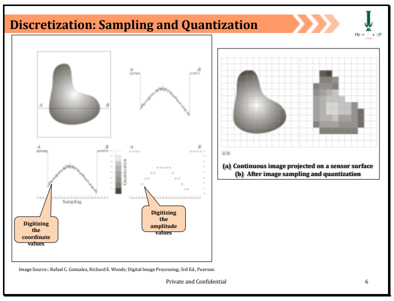


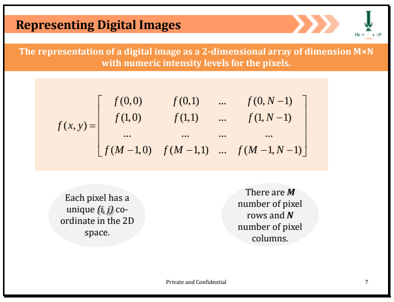
Image formation: Light falls on a source; light is reflected from the surface of the object; the reflected ray is captured by a device or falls on the eyes of a viewer making the object visible.

Slide 6



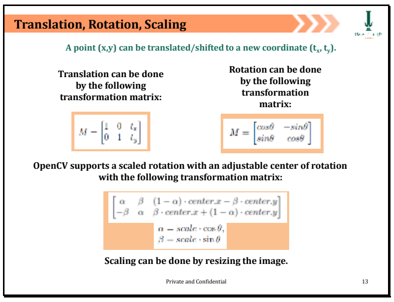
Sampling rate decides how far one sample point is spaced from the adjacent point.
While the quantization levels decide how much information can be encoded in every pixel.

Slide 7



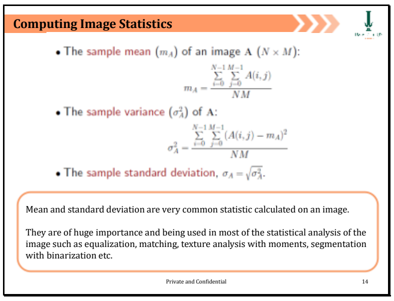
2D matrix representing the spatial extent and coordinates of the pixels. The values of each cell in the matrix represents the intensity value of the corresponding pixel.

Slide 13



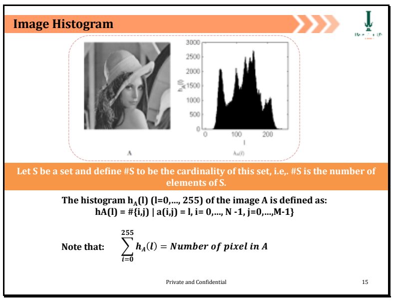
This slide shows the various transformation matrices for different operations – translation, rotations and scaling.

Slide 14



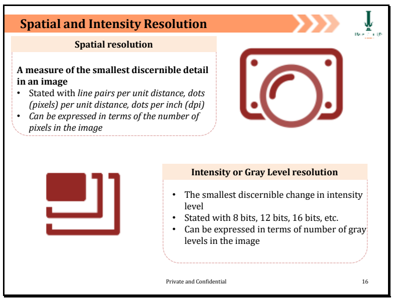
Computing basic statistics from a 2D matrix (image)

Slide 15



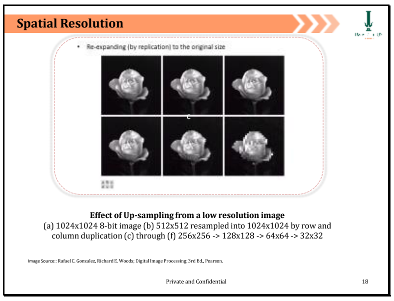
Histogram is the frequency distribution of intensity levels in an image.

Slide 16



Spatial resolution represents the smallest discernible detail in an image. Represented by dpi.
While intensity resolution represents the number of bits per pixel.

Slide 18



Self-explanatory. See the amount of degradation in subsequent images.

Slide 21

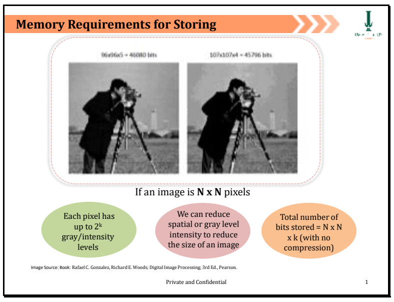
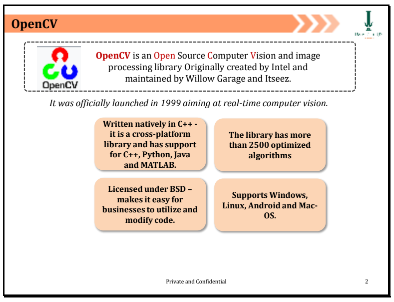


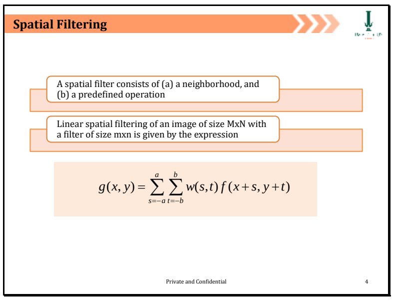
Image size depends on number of bits assigned per pixel.

Slide 23



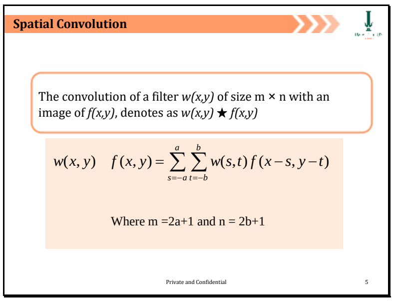
Introduction to OpenCV

Slide 25



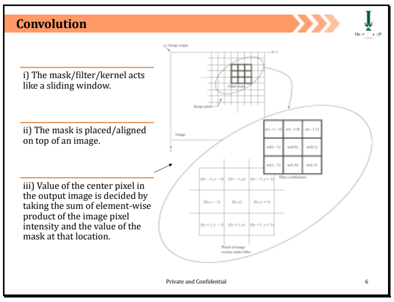
Defining spatial domain filtering mathematically.
The definition of the filter **w(s,t)** will vary for the actual operation (e.g. smoothing, median filter etc.).

Slide 26



Defining convolution: taking dot product of corresponding elements and summing up all the values.
Dot product is the element-wise product of the terms and summing up all such product terms into a single value.
Optionally the values are normalized by dividing the sum of all the values in the filter.

Slide 27



Visually showing the process of convolution with a sliding window (the window represents the convolution filter).

Slide 28

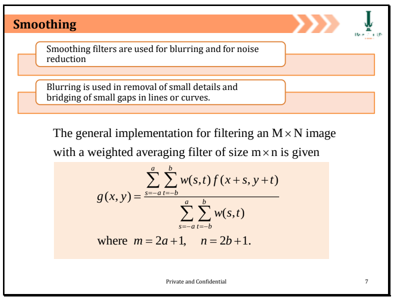


Image smoothing operations with a smoothing filter (e.g. averaging filter).

Slide 29

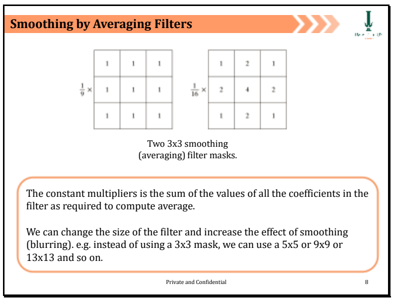
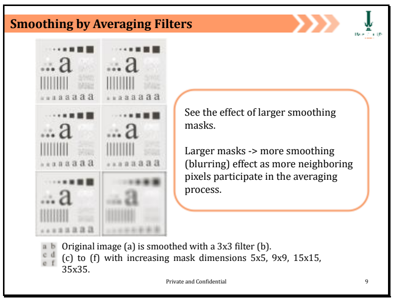


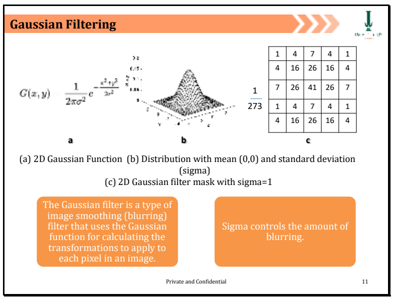
Image smoothing operations with a smoothing filter (e.g. averaging filter).

Slide 30



Results of smoothing (averaging filter).

Slide 32

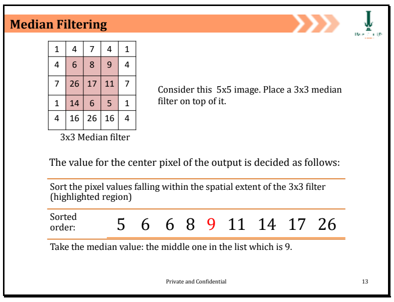


Showing the equation of a Gaussian mask; its plot in 2D; and a representative 5x5 Gaussian filter.

Slide 33

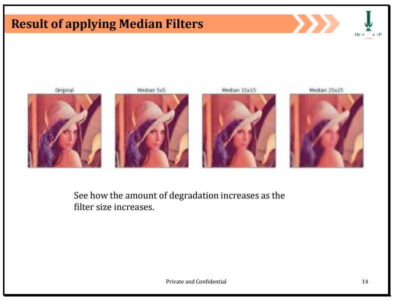
Result of smoothing with a Gaussian filter.

Slide 34



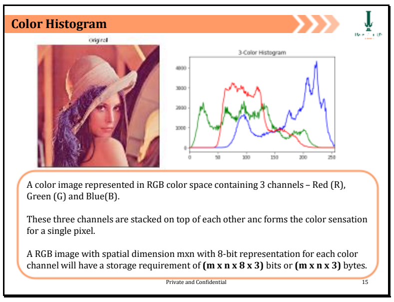
Defining a median filter.

Slide 35



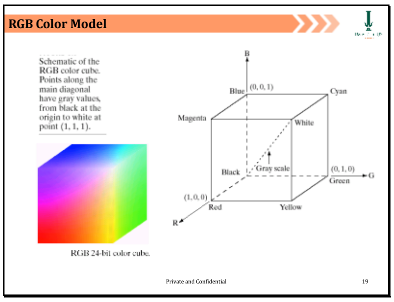
Results of applying a median filter.

Slide 36



Plotting the color histogram – plot of red, green, blue channels of an RGB image.

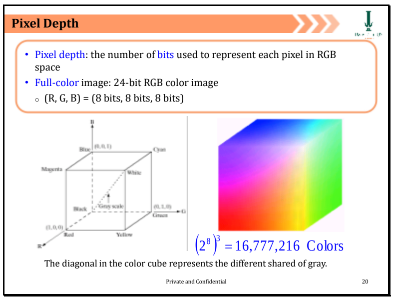
Slide 40



Color cube defining a 3D space of all possible color that is a combination of the Red, Green and Blue color components. Here R, G and B represent three axis in a 3 dimensional coordinate system. We can consider a unit cube and all the colored pixel is a data point in this 3D space.

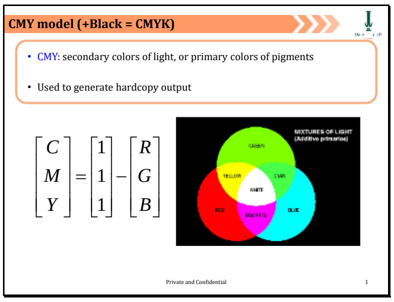
Each pixel is having a specific value for each of the R, G and B component.

Slide 41



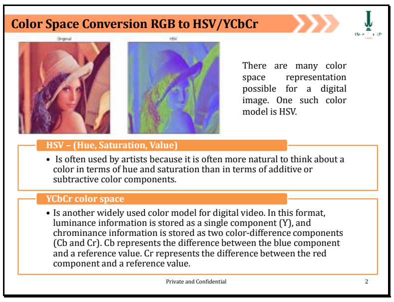
Color cube defining a 3D space of all possible color that is a combination of the Red, Green and Blue color components.
Number of possible color to be represented depends on how many bits are assigned to each pixel for representing each of the basic R, G and B components.

Slide 42



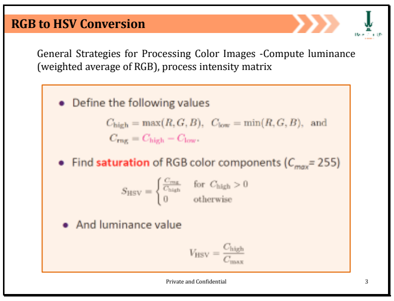
CMY a subtractive color model.

Slide 43



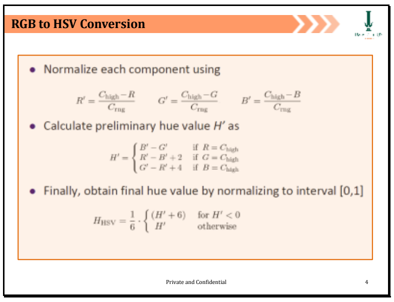
Choosing an RGB image and the same image converted to HSV color-space.

Slide 44



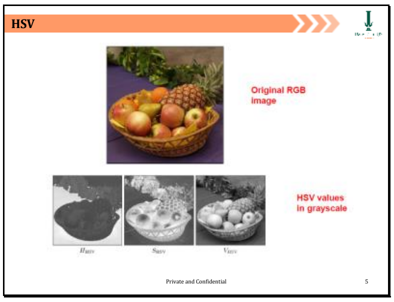
How to convert RGB image to HSV color space?

Slide 45



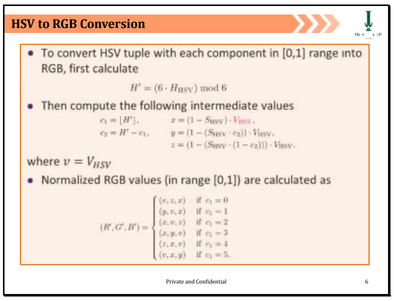
How to convert RGB image to HSV color space?

Slide 46



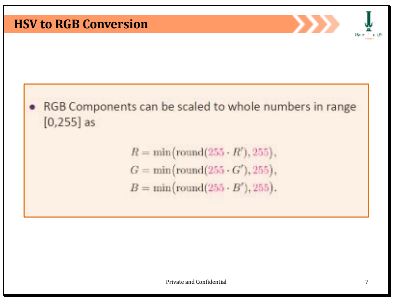
Example

Slide 47



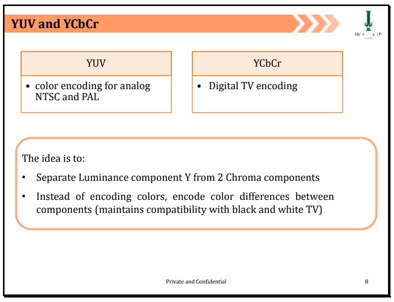
How to convert HSV image to RGB color space?

Slide 48



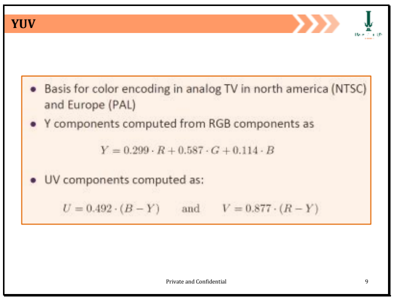
How to convert HSV image to RGB color space?

Slide 49



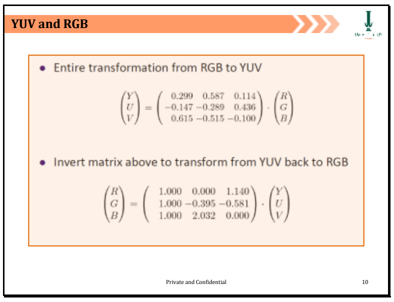
Tow other color models – YUV and YCbCr.

Slide 50



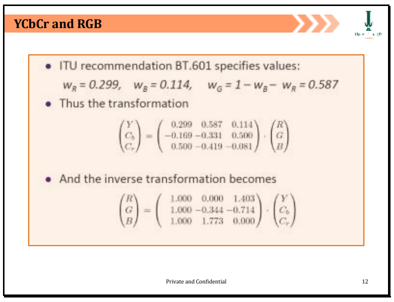
Computing YUV components from RGB.

Slide 51



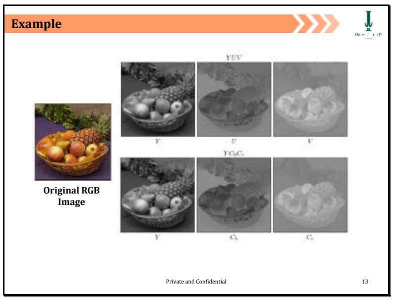
Transformation matrix for conversion between YUV and RGB.

Slide 53



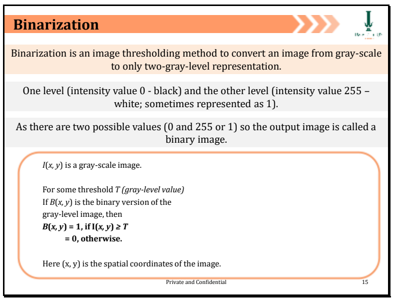
The formula for computing Y is sometimes used to convert a RGB image to grayscale just by considering the Y component.

Slide 54



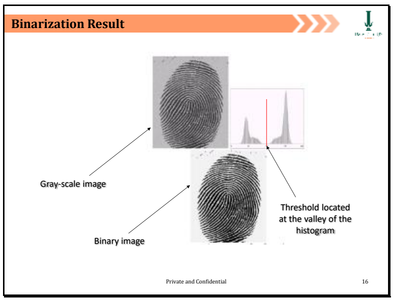
Example: Converting a RGB image to YUV and YCbCr.

Slide 56



Algorithm for Binarization.

Slide 57

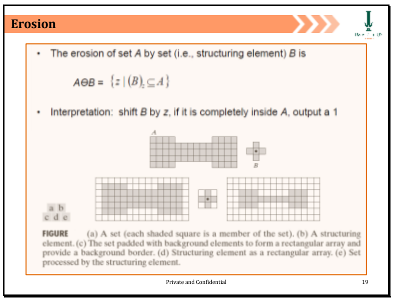


Example of image binarization.

Slide 58

Results of binarization.

Slide 60



Explaining different binary morphological operations.

Slide 72

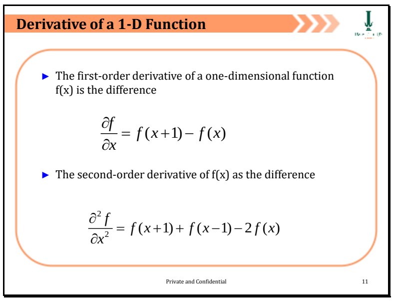
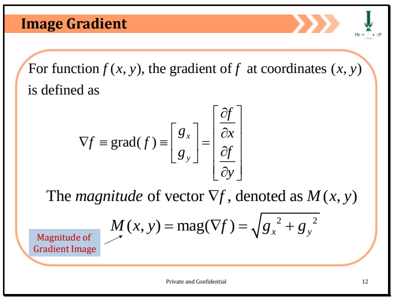


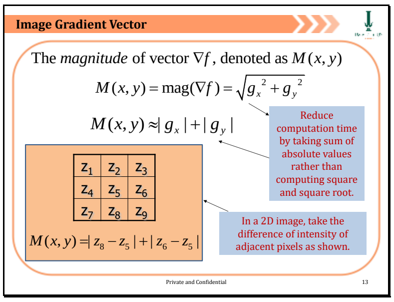
Image derivate approximated with difference of intensity values of consecutive pixels.

Slide 73



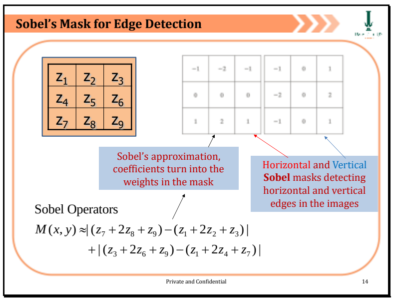
Defining image gradient vector and its magnitude.
g_x is the derivative of the image w.r.t x
g_y is the derivative of the image w.r.t y
grad(f) is the vector containing the g_x and g_y as component.
M(x,y) is the magnitude of the vector. Standard Euclidean norm formula.

Slide 74



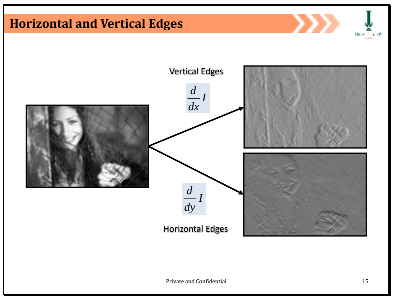
Approximating image gradient with difference of adjacent pixel-intensities in a 3x3 image region.
Refer to the previous slide for a definition of gradient.

Slide 75



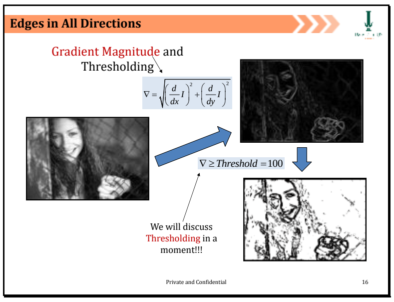
Approximating image gradient with difference of adjacent pixel-intensities in a 3x3 image region.

Slide 76



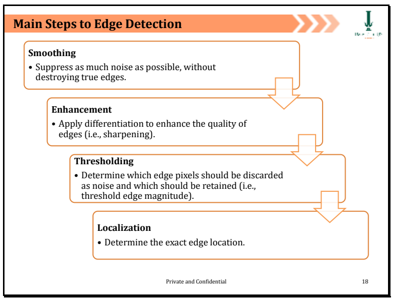
Results of image gradients – derivative w.r.t x and y.

Slide 77



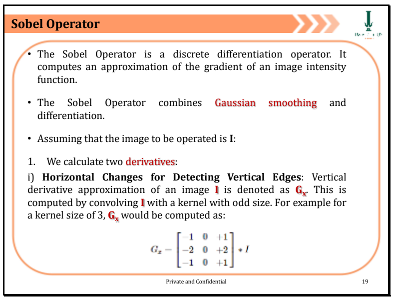
Results of image gradients – derivative w.r.t x and y. Taking the magnitude of the gradient vector.

Slide 79



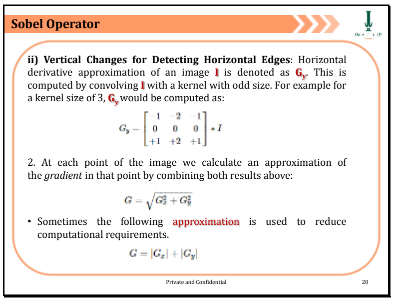
Algorithm for edge detection.

Slide 80



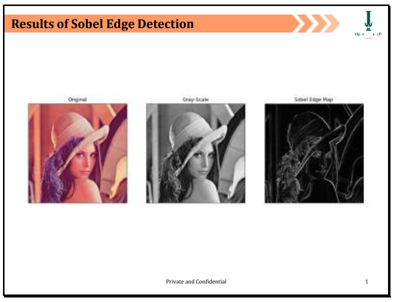
Defining the Sobel's operator for edge detection.

Slide 81



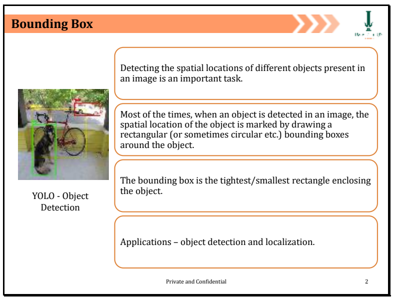
Defining the Sobel's operator for edge detection.
Gradient may be approximated by taking the sum of absolute values of the gradients computed in the X and Y directions in order to reduce computational complexity.

Slide 82



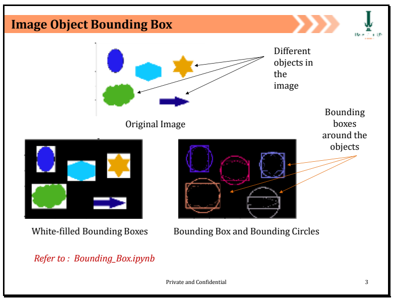
Results of edge detection.

Slide 83



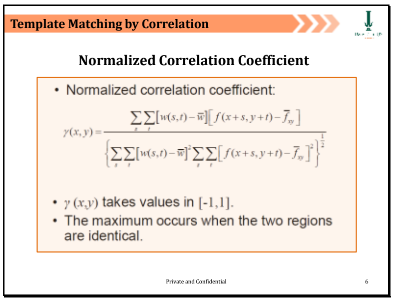
Object detection from images – demarcation of spatial extents of individual object with a suitable bounding box.

Slide 84



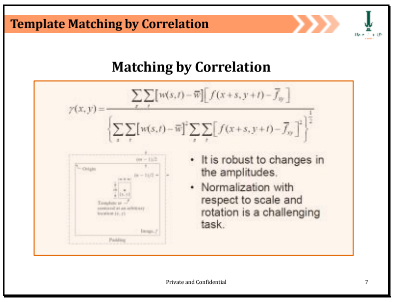
Detecting and drawing bounding boxes around each object in the image.

Slide 88



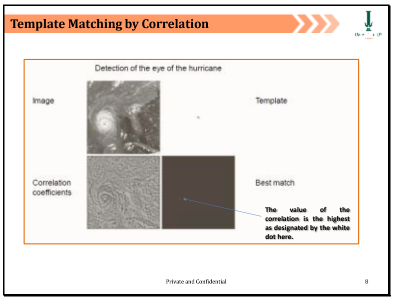
Computing normalized correlation coefficient for image matching.
w(.) is the sliding window – representing an image patch with which matching is done – template image.
f() is the original base image. We are looking for a match in the base image.
f(x+s, y+t) is the part of the image under the window w(s,t) at a given time.
w_bar is the average of the intensities of the pixels inside a given window – template image.
f_{xy}_bar is the average of the intensities of the pixels of the base image falling inside a given window.

Slide 89



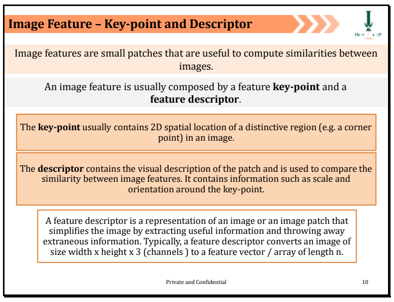
Computing normalized correlation coefficient for image matching.
w(.) is the sliding window – representing an image patch with which matching is done – template image.
f() is the original base image. We are looking for a match in the base image.
f(x+s, y+t) is the part of the image under the window w(s,t) at a given time.
w_bar is the average of the intensities of the pixels inside a given window – template image.
f_{xy}_bar is the average of the intensities of the pixels of the base image falling inside a given window.

Slide 90



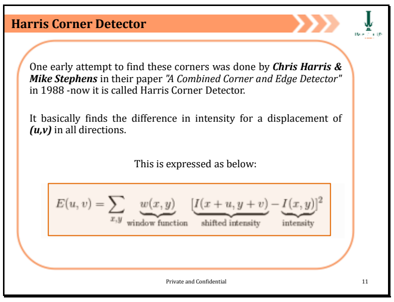
Example of matching by correlation.
This is showing the result of the operation stated in the previous slide.

Slide 92



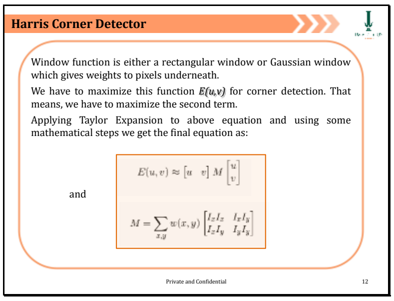
Defining image feature – key-point in an image and key-point descriptor.

Slide 93



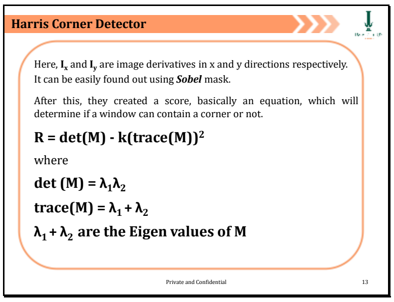
Find the difference in intensity displacement in the image.

Slide 94



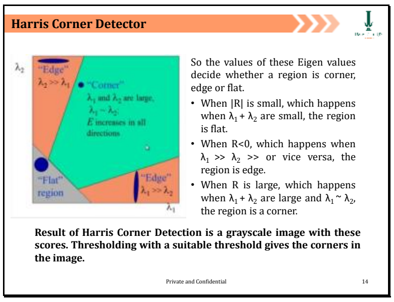
Maximizing E(u,v) and computing gradients.

Slide 95



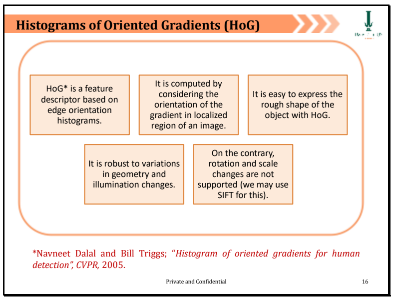
Determine the Eigen values.
Determinant is a single value that can be computed from a square non-singular matrix.
Trace of a square matrix is the summation of all the elements in the main diagonal.
These are standard linear algebra terms. Students are expected to have some background of linear algebra and calculus (10+2 maths.)

Slide 96



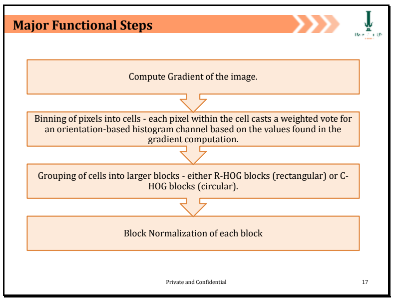
Make decisions whether a given point is a corner or not.

Slide 98



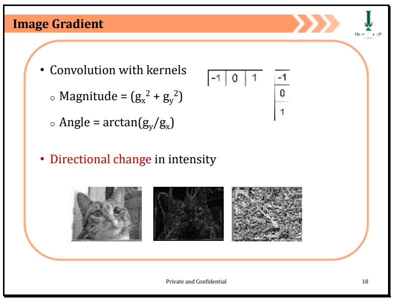
HoG: Original paper was published in 2005 and used for human detection. Major steps involved in HoG are stated in the subsequent slides.

Slide 99



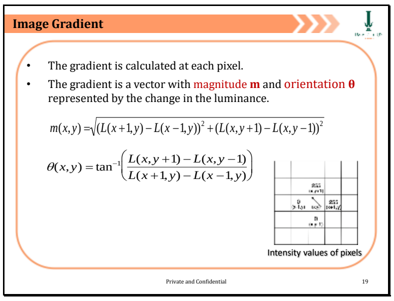
Major steps involved in HoG.

Slide 100



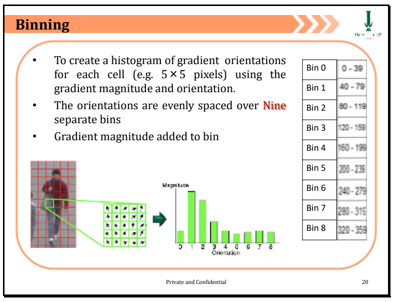
Computing image gradient.

Slide 101



Computing image gradient – magnitude and orientation direction w.r.t a pixel located at (x,y).

Slide 102



Gradient orientation binning.

Slide 103

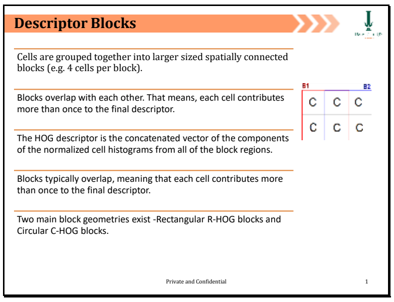
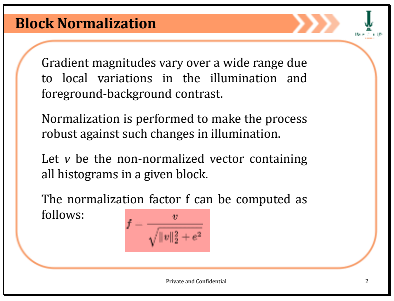


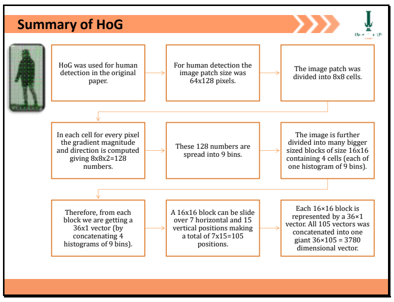
Image is divided into larger size blocks. Each block is divided into cells. Breaking down the image into such different sized spatial-extents ensure better invariance to changes that may happen during capture of the image.

Slide 104



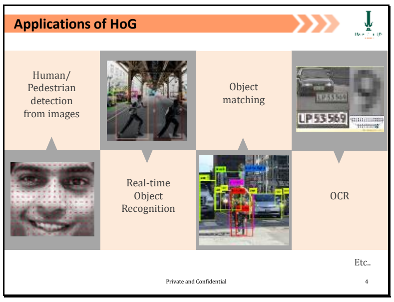
Block normalization reduces the effect of illumination changes in the surroundings. The process of normalization reduces the effect of lighting changes in the scene.

Slide 105



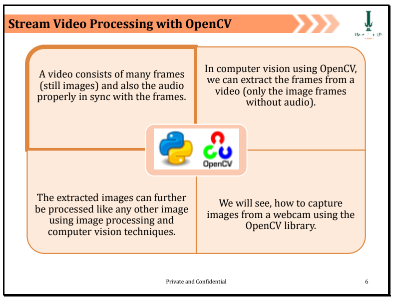
Summarizing the whole process for a human detection case.

Slide 106



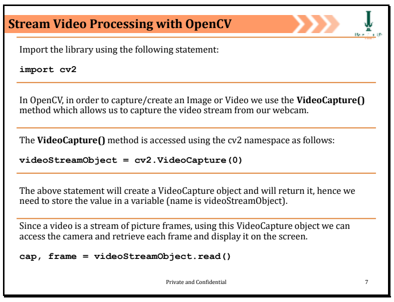
Applications of HoG

Slide 108



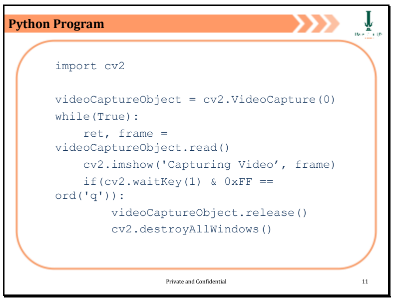
What is a video stream?

Slide 109



Step-by-step description of how to extract frame from a video stream captured by the webcam.

Slide 113



The complete Python code for extracting frames from the video captured by a webcam.