Digital Image Processing

Lecture 8
Object Recognition 2

Rahul JAIN

Spring 2022

Outline

- SIFT Descriptor
- Histogram of Oriented Gradients

SIFT Detector

- Interest Points
- Detecting Blobs (Binary Local Object)
- SIFT Detector
- SIFT Descriptor

- Size
- Orientation
- Lighting
- Object occlusion

SIFT Detector

- Interesting points (Keypoints):
  - Rich image content (brightness variation, color variation etc)
  - Signature (representation) for matching/comparing with other points
  - Well-defined position
  - Invariant to image rotation and scaling

SIFT Detector

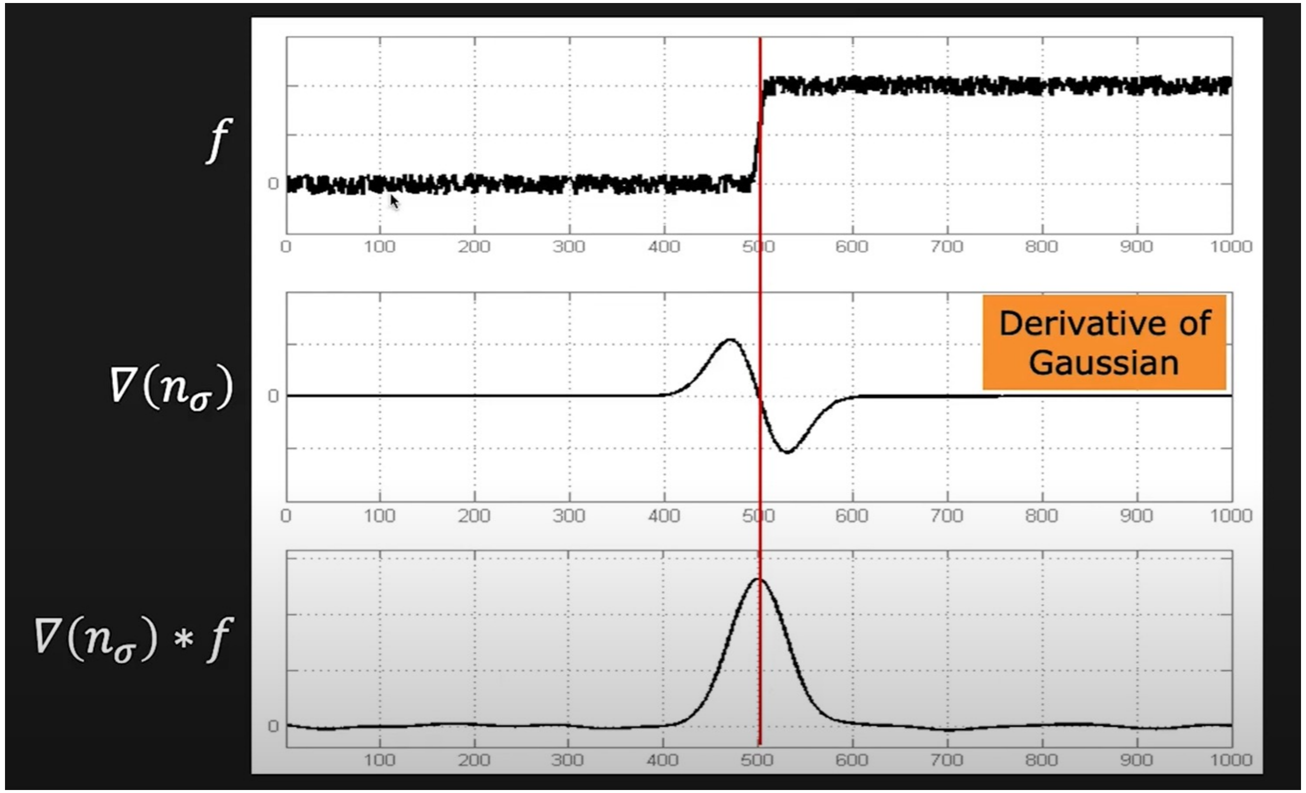
- Locate the keypoint
- Determine the size
- Determine the orientation
- Formulate a description/signature (independent of size and orientation)

SIFT Detector

- Gaussian operator for 1-*D* function

First derivative with respect to x, of the Gaussian

Convolve the image with Gaussian

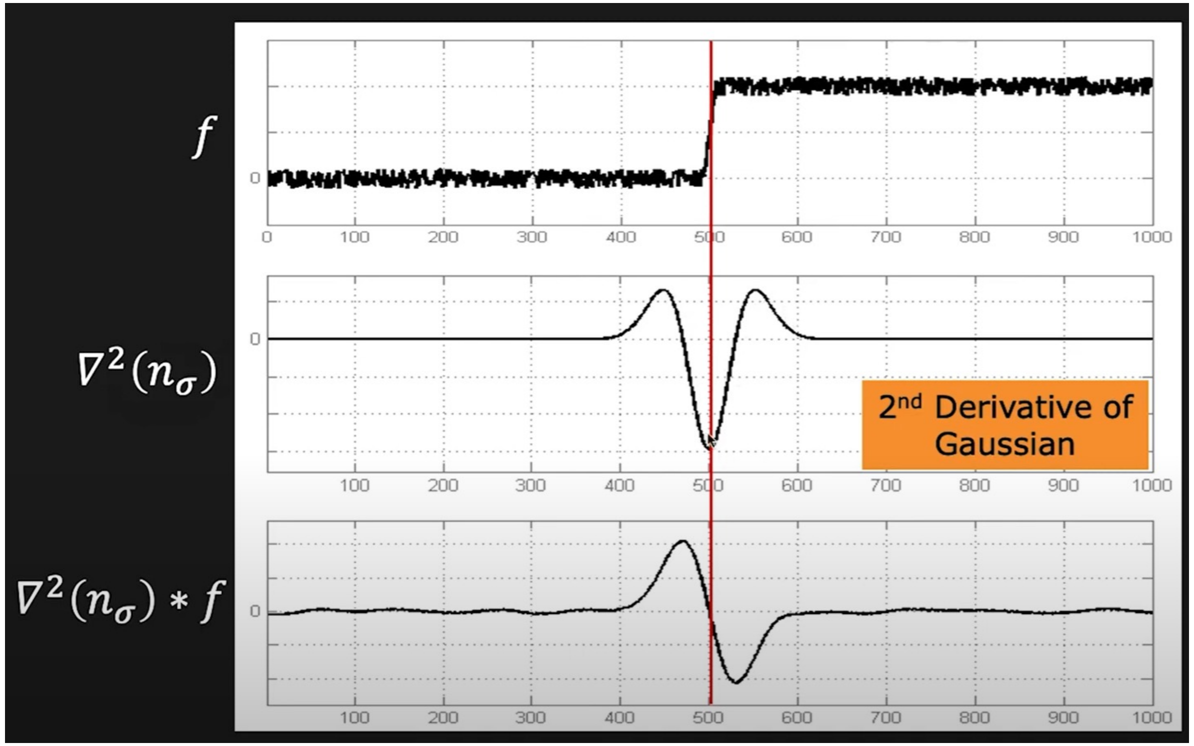


Extremum of Derivative of Gaussian denotes an Edge

Source: All image are taken form https://fpcv.cs.columbia.edu/

SIFT Detector

- You can find the edge using second order derivative



Zero crossing in 2nd Derivative of Gaussian denote an Edge

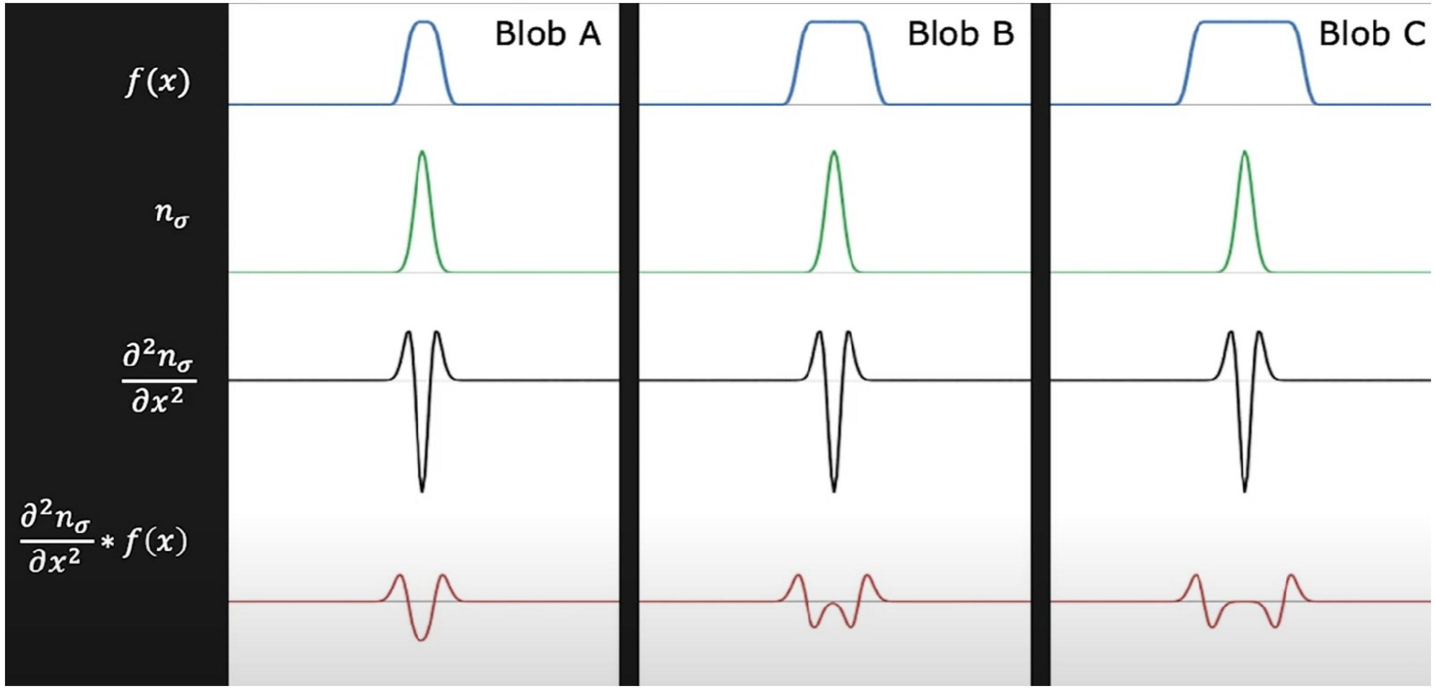
SIFT Detector

Different sizes of Blob

Gaussian filter with *n*

Second order derivative of Gaussian

Convolve the image with Gaussian to get the peak (Edge)



SIFT Detector

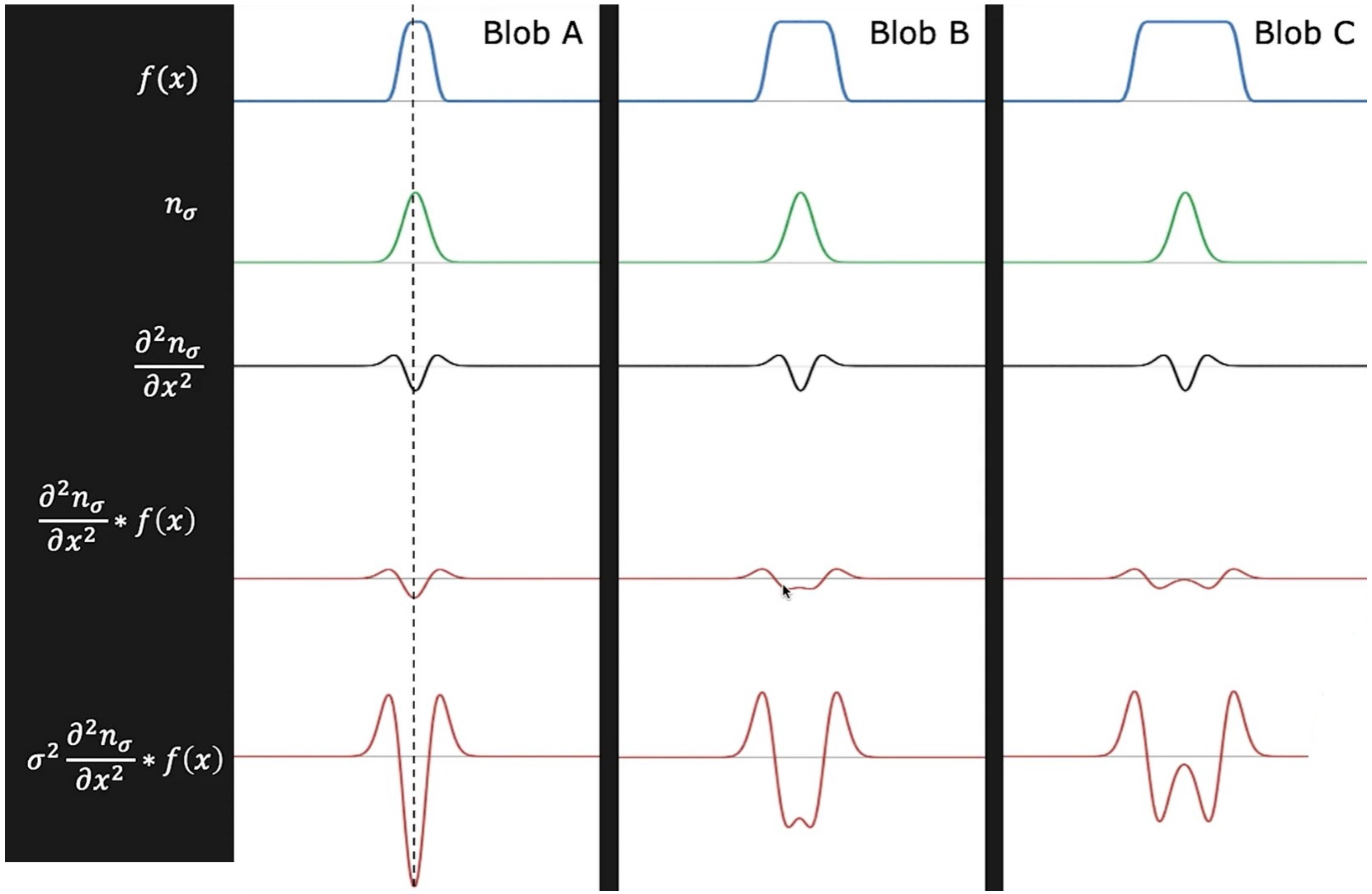
Different sizes of Blob

Gaussian filter with *n*

Second order derivative of Gaussian

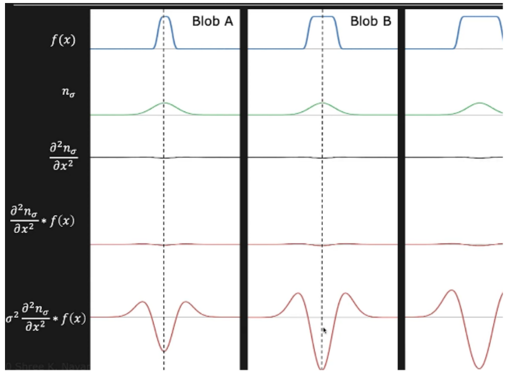
Convolve the image with Gaussian to get the peak (Edge)

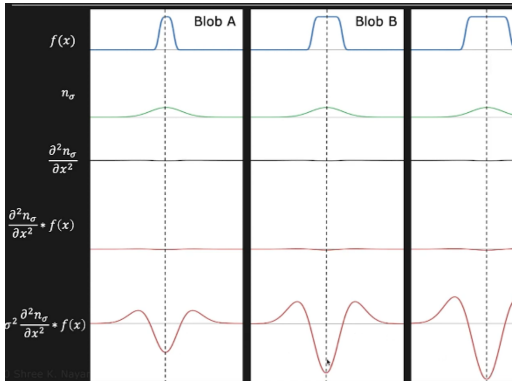
Normalized: Second order derivates of Gaussian with sigma squared



SIFT Detector

- Normalization:
  - The response of the Gaussian operator will depend on the sigma that we use
  - As a sigma gets wider its peak value begins to fall
  - The response of the operator also reduces with sigma
- When we increase the value of sigma the response is distinct peak





SIFT Detector

- Apply the second derivative of Gaussian for multiple sigmas
- Apply a large number of sigmas, sigmas is referred to as scale
- Irrespective of size of the blob, we can produce some a maximum that the location of blob at some scale
- After that, we have stack of images which corresponding to different scales that is the output of the second derivate of the Gaussian applied to the image at multiple sigma values
- Local Extrema in (x, σ) space represent blobs

σ – scale, x – spatial coordinates

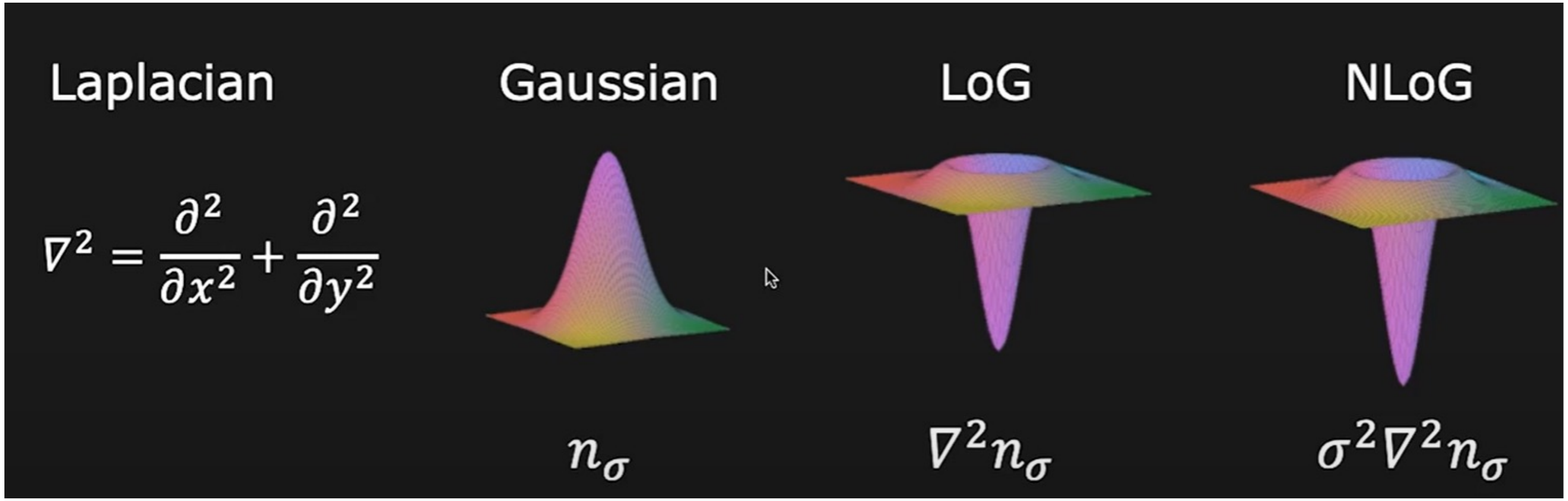
SIFT Detector

- For given $f(x)$
- Compute: $\sigma^2 \frac{\partial^2 n_\sigma}{\partial x^2} \ast f(x)$ at many scales (σ)
- In stack, we will find the Extremum corresponding to the position and scale

arg max (x, σ)

SIFT Detector

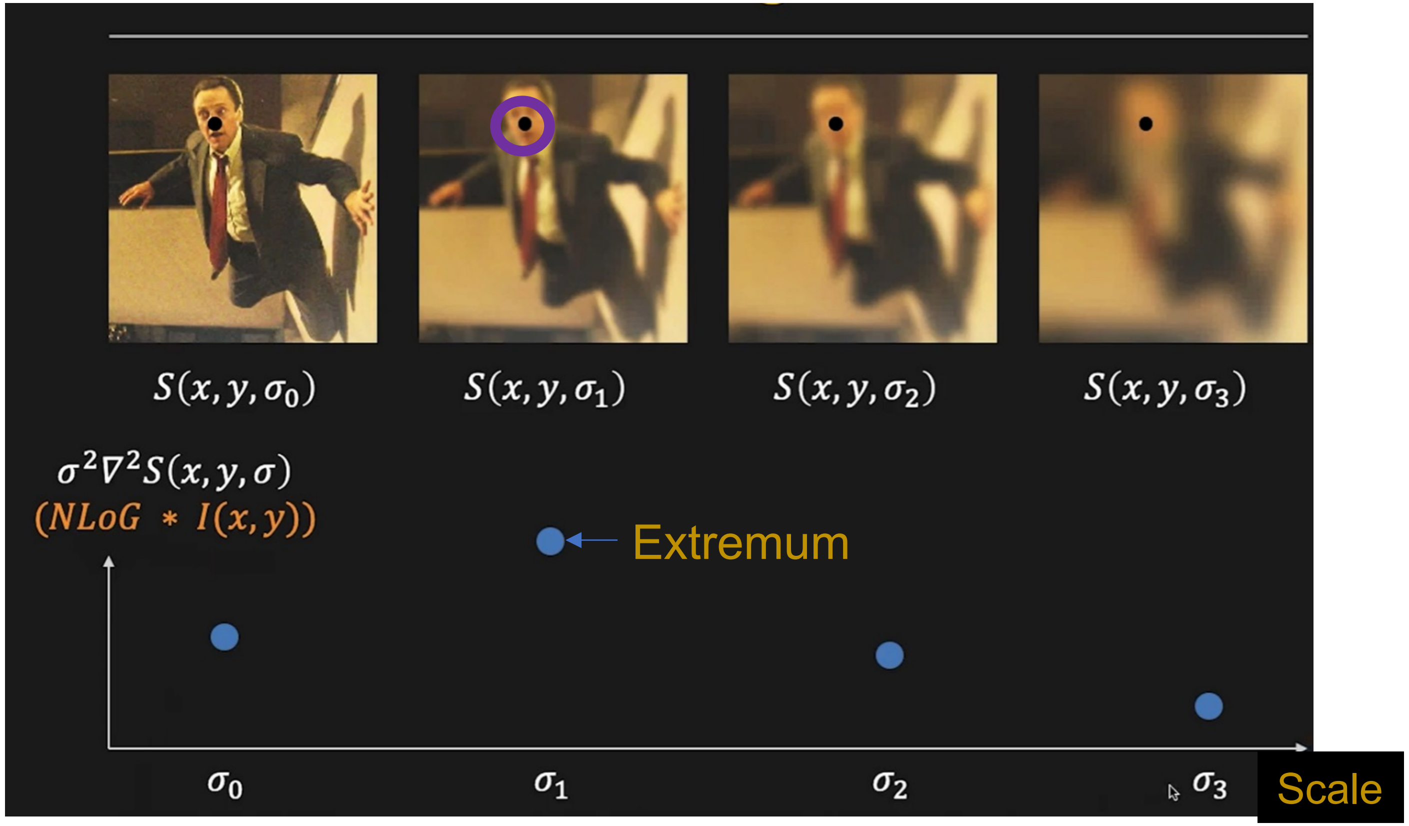
- For 2D image: Normalized Laplacian of Gaussian is used as the 2D image



- Location of Blobs given by Local Extrema after applying Normalized Laplacian of Gaussian at many scales
- Apply Gaussian filter with multiple sigma values to get a stack of volume

SIFT Detector

- Stack Space: Stack created by filtering an image with Gaussian of different sigma (σ)



SIFT Detector

- Given an image $I(x, y)$
- Convolve the image suing NLoG at many scales σ

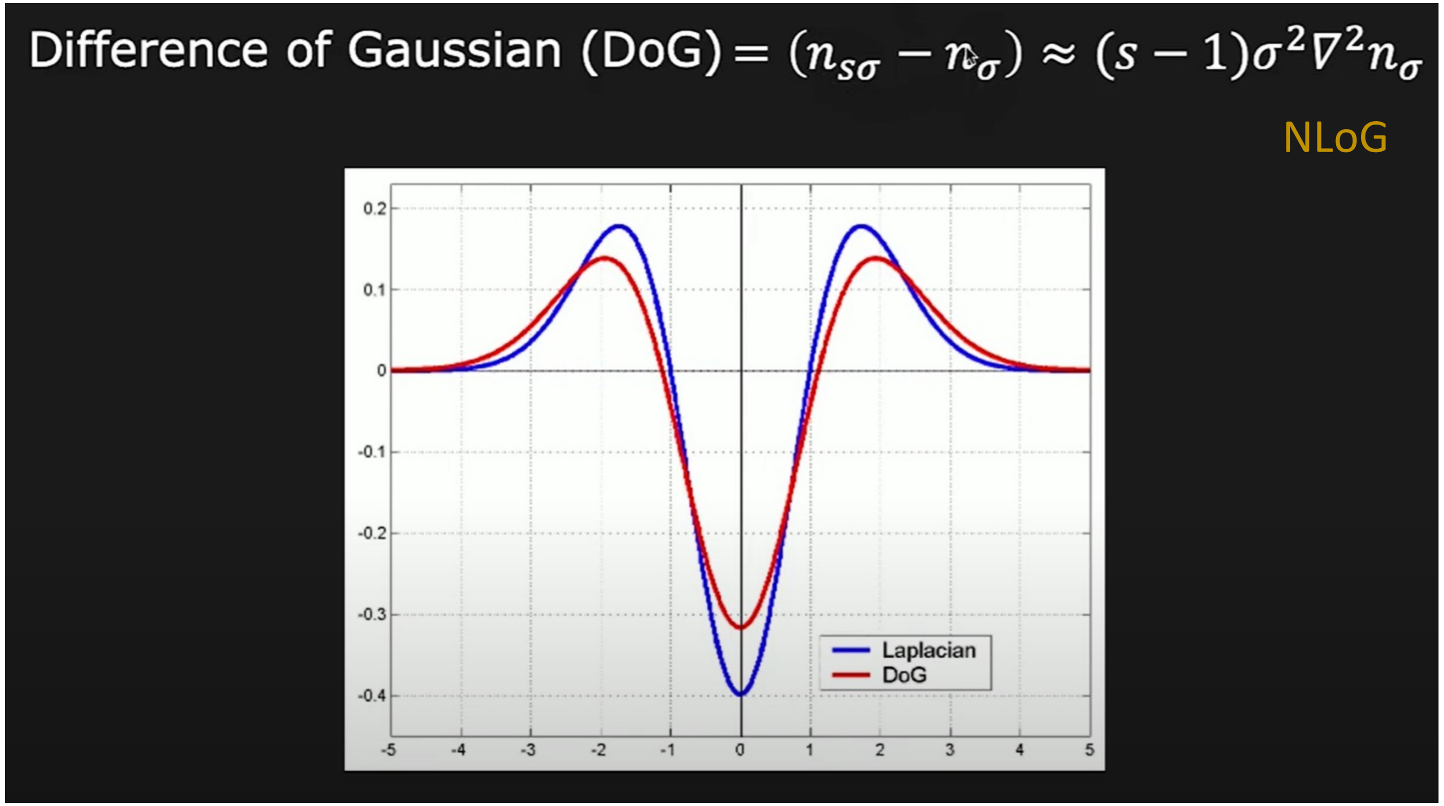
$$(x^*, y^*, \sigma^*) = \underset{(x,y,\sigma)}{\arg max} |\sigma^2 \nabla^2 n_\sigma \; * I(x, y)|$$

(x^*, y^*) = Position of the blob

σ^* = Size the blob

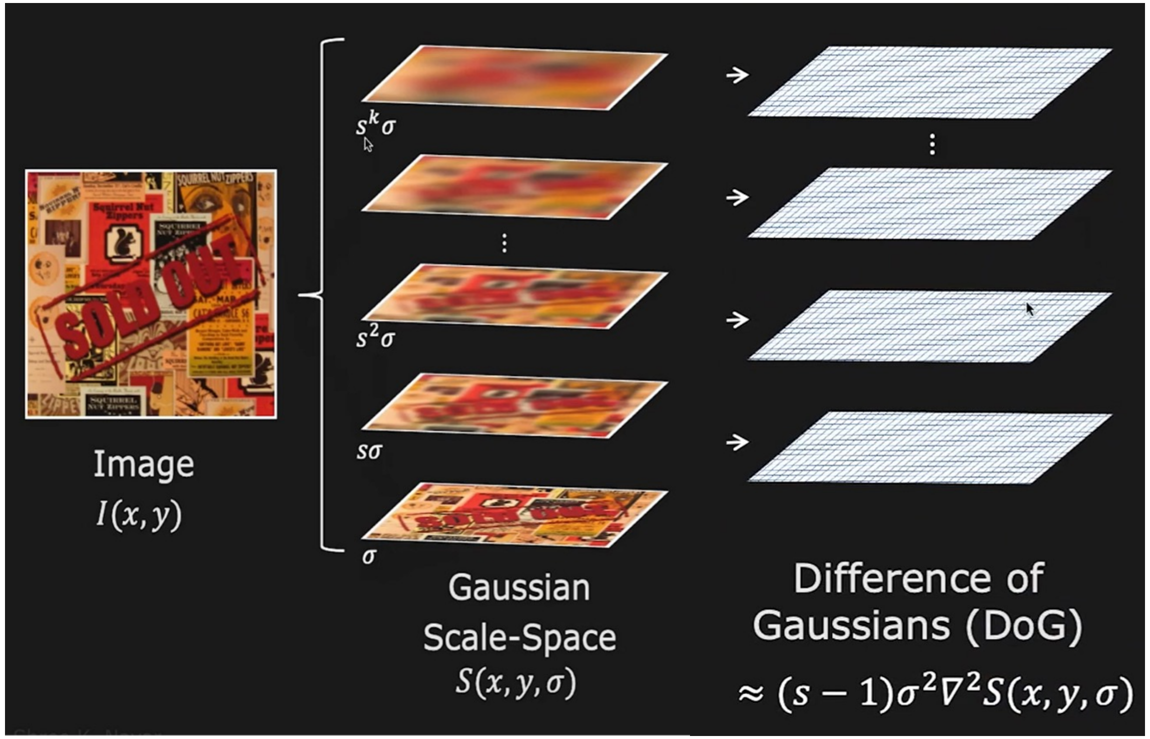
SIFT Detector

- Difference of Gaussian (DoG) : Enhance the information



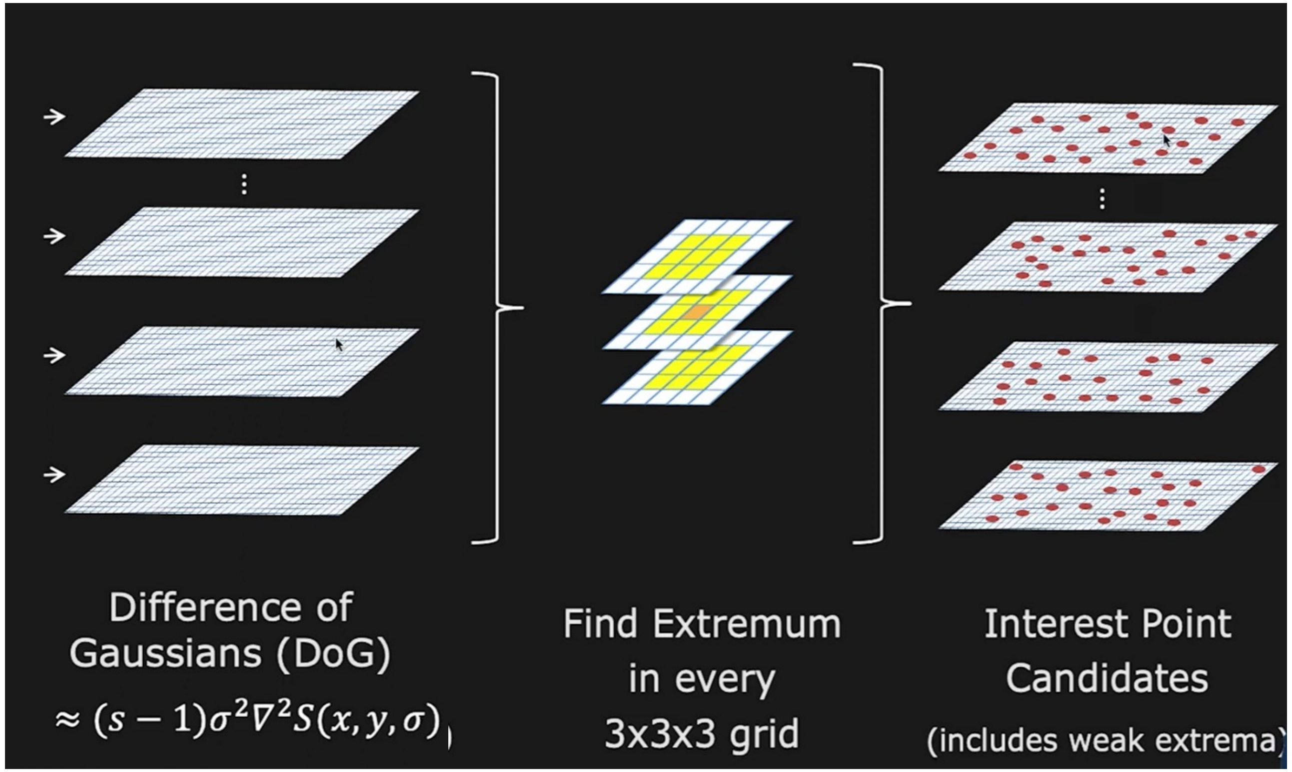
SIFT Detector

- Find keypoints



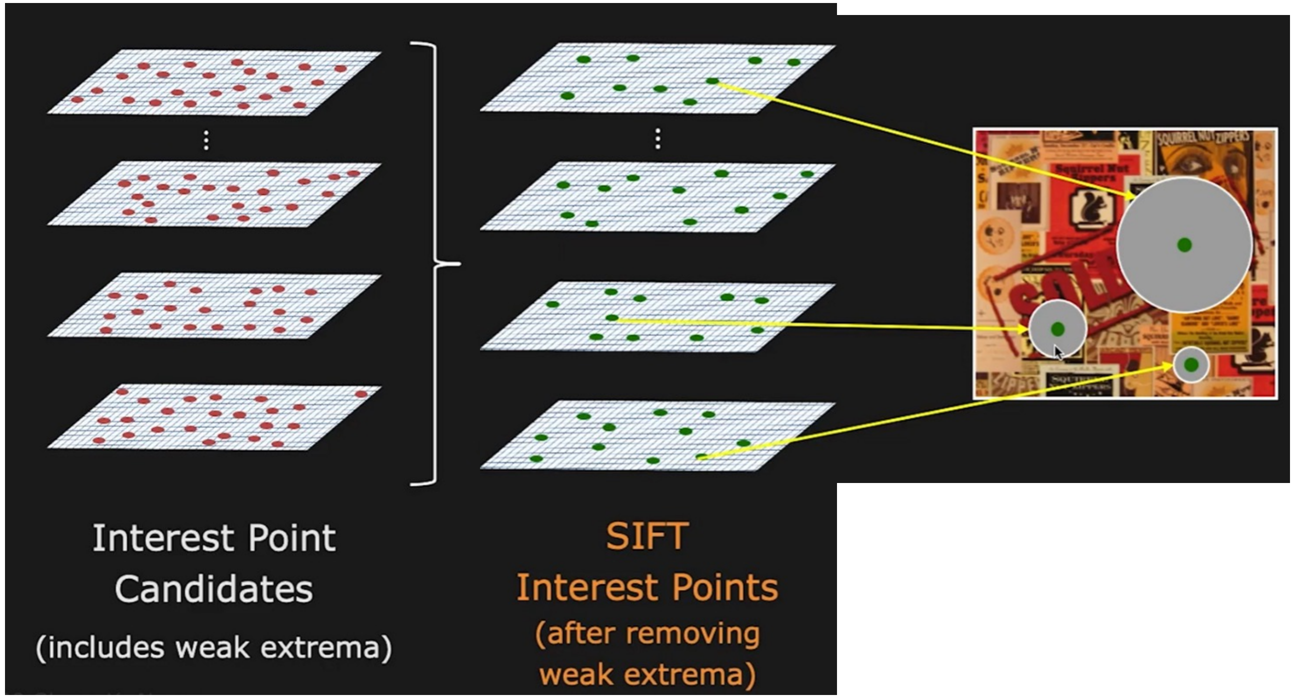
SIFT Detector

- Keypoints



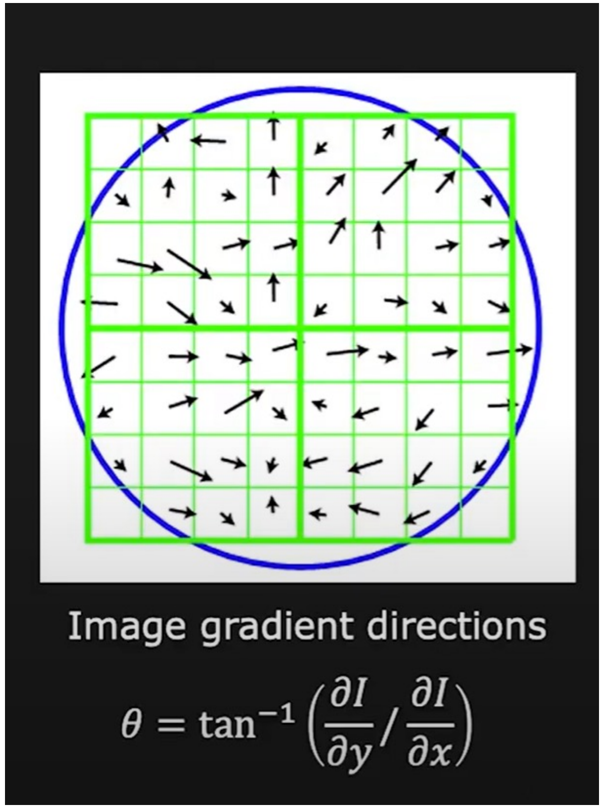
SIFT Detector

- Elimination of the bad keypoints: Threshold, Hessian determinant



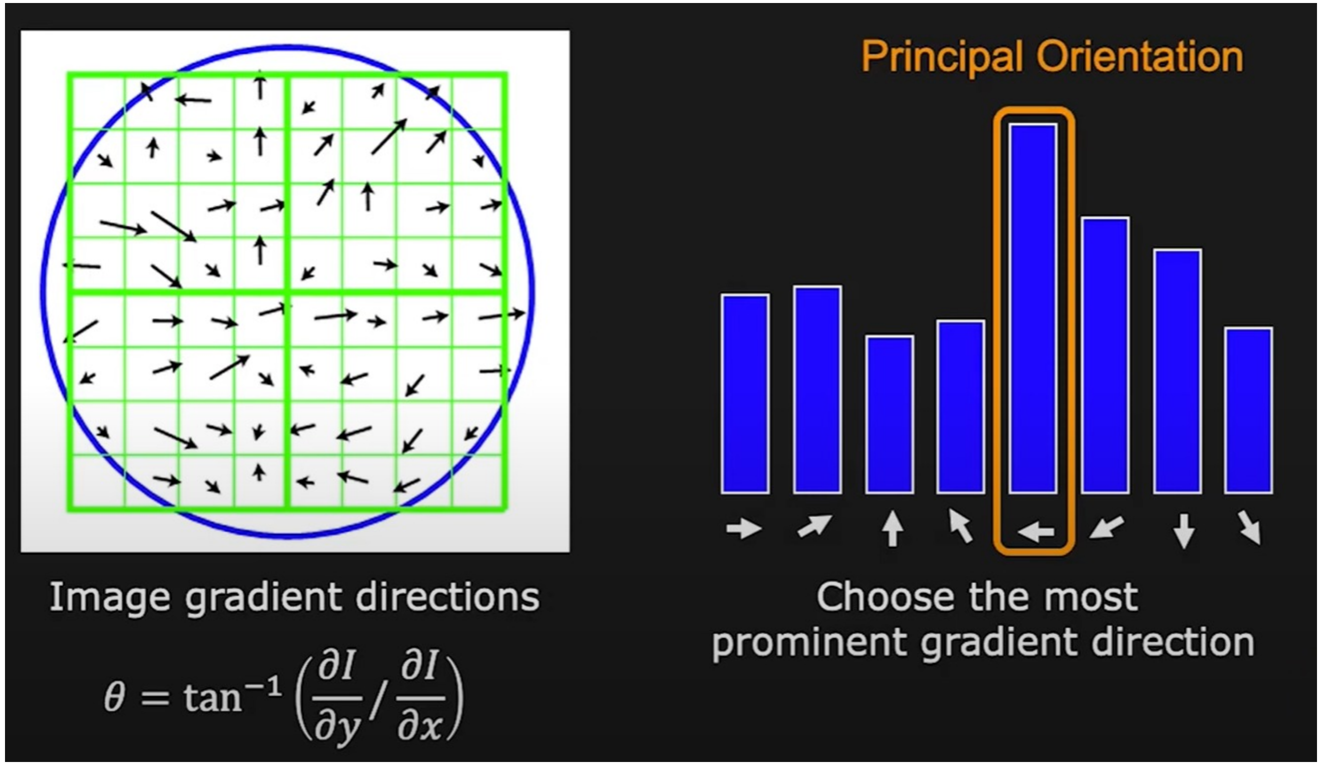
SIFT Detector

- Computing the principal orientation
- Use the histogram of gradient directions



- After the scale has been removed this is area that the blob occupied
- We can place a grid in there
- Every pixel in the grid we can compute the gradient
- Apply gradient operator which gives orientation of the edge and magnitude
- Ignore magnitude (Magnitude is affected by lighting)
- Orientation Direction of the gradient is taken
- Create a histogram

SIFT Detector

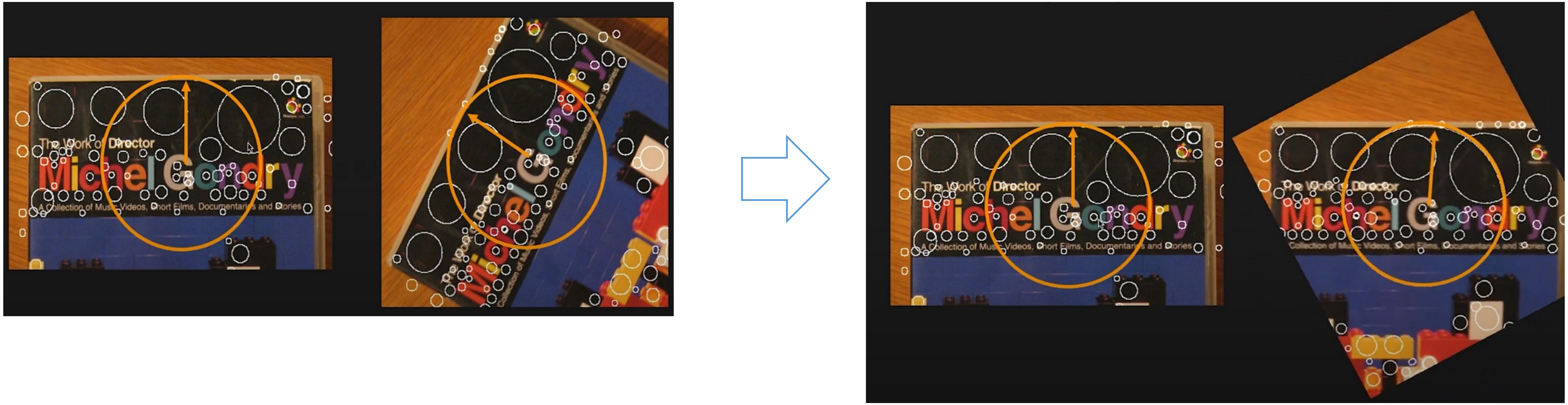


Rotation Invarience

- Create a histogram
- X-axis represents direction and bars corresponding to the number of pixels within the region, which have that particular direction (edge direction)

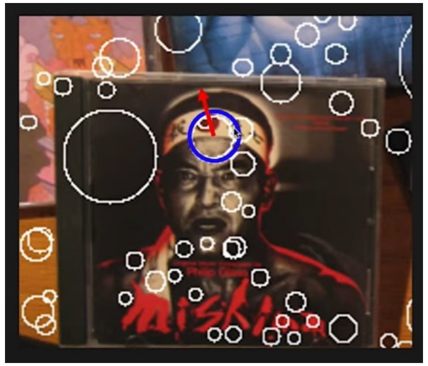
SIFT Detector

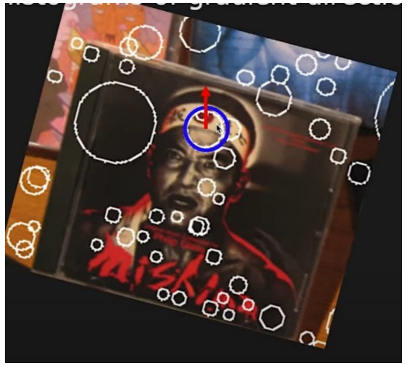
- You can compute orientation and undo the rotation of one to match the other

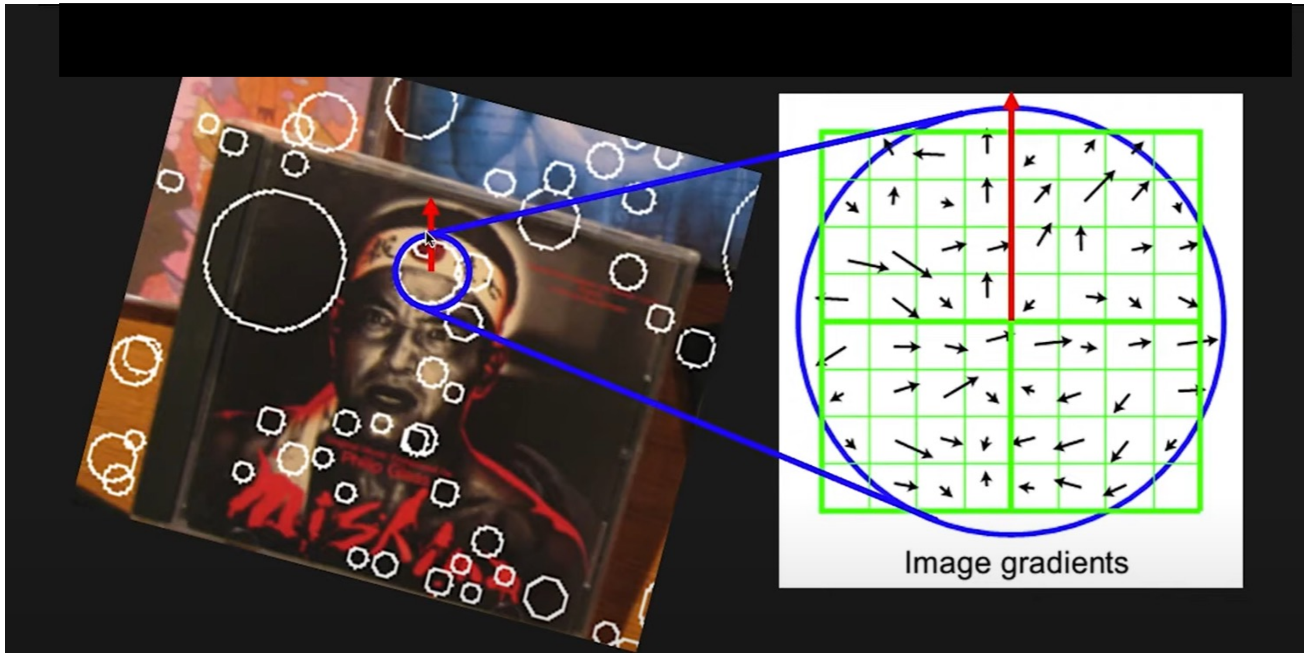


SIFT Descriptor

- Histograms of the gradient directions over spatial regions

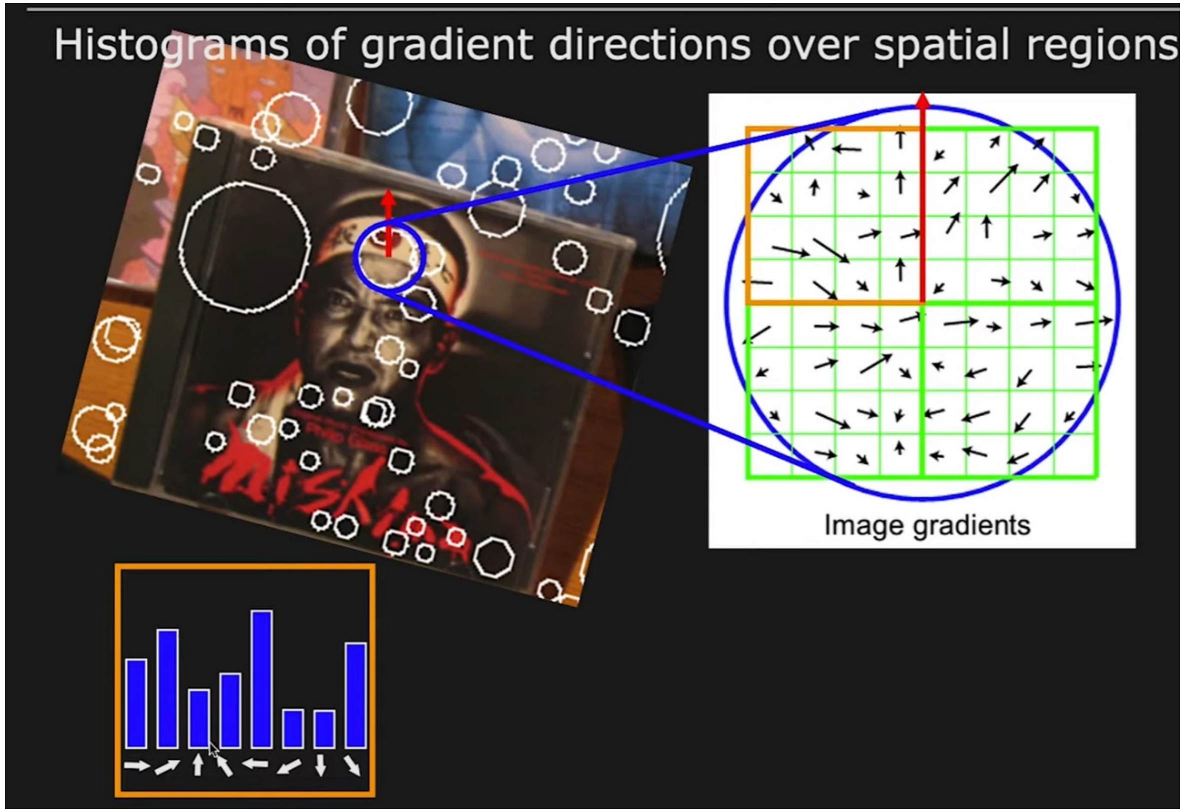






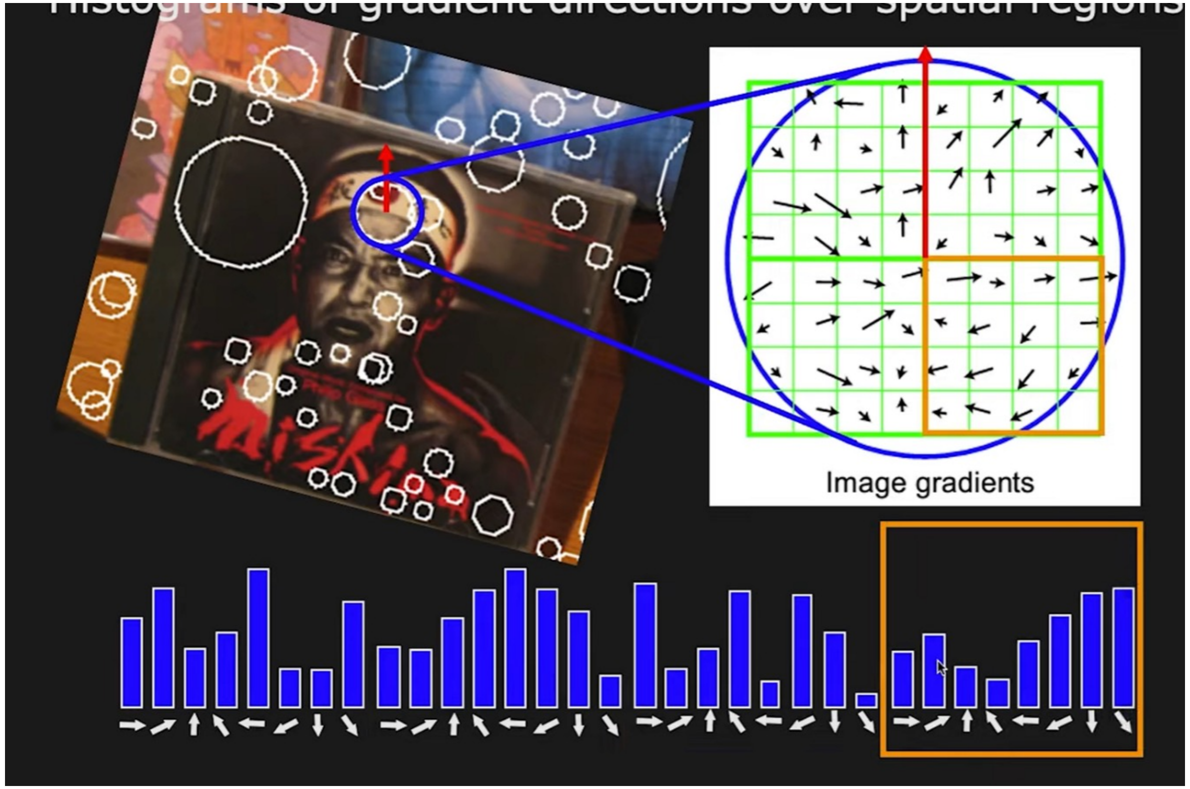
SIFT Descriptor

- We place some standard size of grid here
- We can compute gradient of each pixel (magnitude and orientation)



SIFT Descriptor

- Normalized histogram: Invariant to Rotation, Scale, Brightness
- We can use as a descriptor signature for matching to SIFT features



SIFT Descriptor

- Comparing SIFT descriptors
- L2 distance
- If $H_1(K)$ and $H_2(K)$ are two arrays of data of length *N*

$$d(H_1, H_2) = \sqrt{\sum_k (H_1(K) - H_2(K))^2}$$

- Normalized correlation (Template matching)
- Intersection matrics

HOG: Histogram of Oriented Gradients

HOG: Histogram of Oriented Gradients

- Approach to extract feature from image retaining crucial features
- It is like Canny edge detector, SIFT (Scale Invariant and Feature Transform)
- It is used for object detection
- The technique counts occurrences of gradient orientation in the localized portion of an image

Histogram of Oriented Gradients

- The HOG descriptor focuses on the structure or the shape of an object.
- It uses both magnitude and orientation of the gradient to compute features.
- For the regions of the image, it generates histograms using the magnitude and orientations of the gradient.

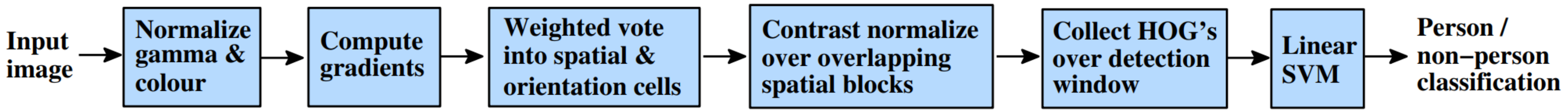


Figure 1. An overview of our feature extraction and object detection chain. The detector window is tiled with a grid of overlapping blocks in which Histogram of Oriented Gradient feature vectors are extracted. The combined vectors are fed to a linear SVM for object/non-object classification. The detection window is scanned across the image at all positions and scales, and conventional non-maximum suppression is run on the output pyramid to detect object instances, but this paper concentrates on the feature extraction process.

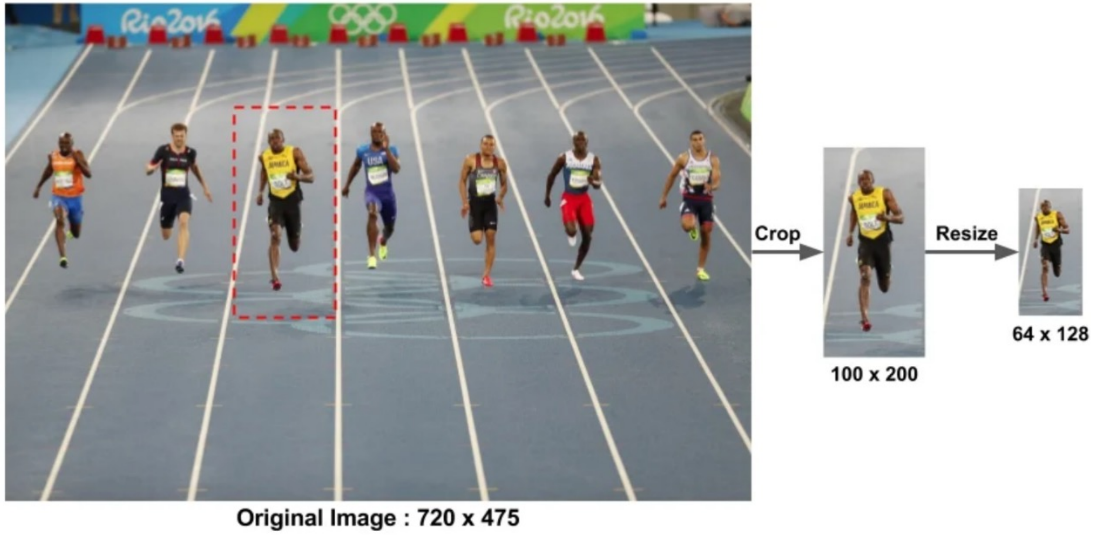
Source: Navneet Dalal and Bill Triggs: "Histograms of Oriented Gradients for Human Detection" CVPR-2005

Histogram of Oriented Gradients

- Steps to calculate HOG Features

Step 1 : Preprocessing

- Resize the input grayscale image into 128 $\times$ 64 (used in the author paper)



Source: https://learnopencv.com/histogram-of-oriented-gradients/

Histogram of Oriented Gradients

Step 2 : Calculate the Gradient Images

- Calculate the horizontal and vertical gradients
- We can find the magnitude and direction of gradient

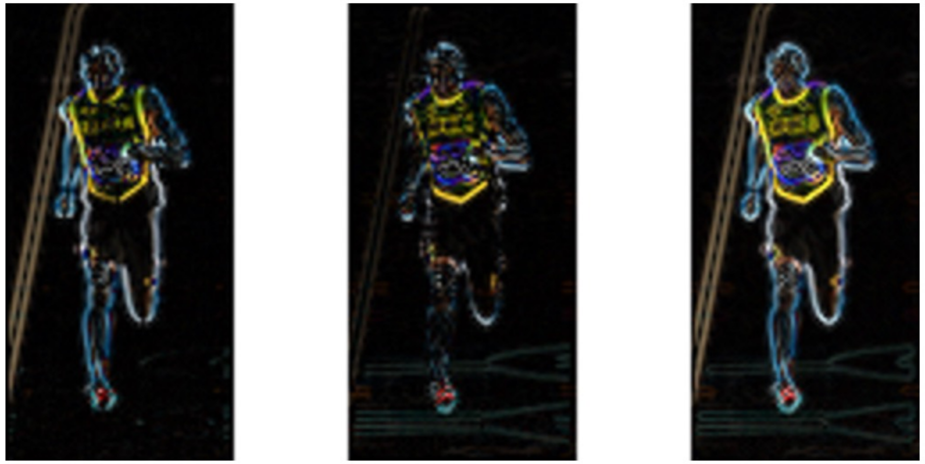
-1	0	1

-1
0
1

Sobel operator with kernel size 1

$$g = \sqrt{g_x^2 + g_y^2} \qquad \theta = \arctan {}^{g_y}/_{g_x}$$

At every pixel, the gradient has a magnitude and a direction. For color images,
the gradients of the three channels are evaluated (as shown in the figure above).
The magnitude of gradient at a pixel is the maximum of the magnitude of gradients of the three channels,
and the angle is the angle corresponding to the maximum gradient.



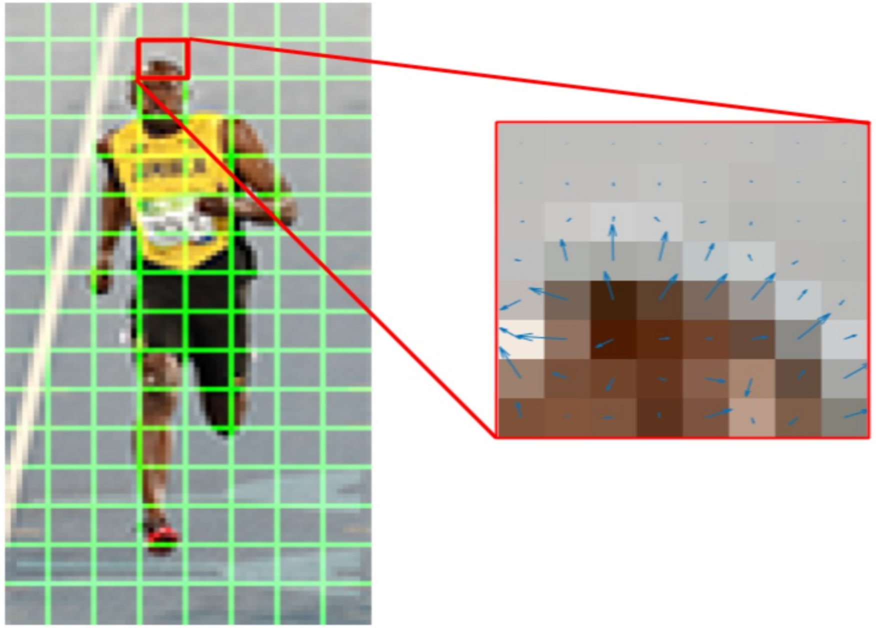
Left : Absolute value of x-gradient. Center : Absolute value of y-gradient. Right : Magnitude of gradient.

Source: https://learnopencv.com/histogram-of-oriented-gradients/

Histogram of Oriented Gradients

Step 3 : Calculate Histogram of Gradients in 8×8 cells

- image is divided into 8×8 cells and a histogram of gradients is calculated for each 8×8 cells



64 x 128

2	3	4	4	3	4	2	2
5	11	17	13	7	9	3	4
11	21	23	27	22	17	4	6
23	99	165	135	85	32	26	2
91	155	133	136	144	152	57	28
98	196	76	38	26	60	170	51
165	60	60	27	77	85	43	136
71	13	34	23	108	27	48	110

Gradient Magnitude

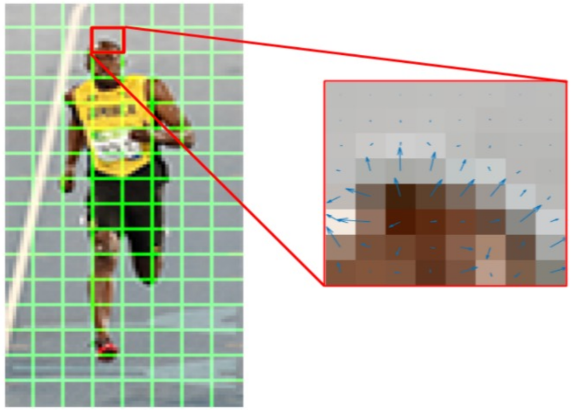
80	36	5	10	0	64	90	73
37	9	9	179	78	27	169	166
87	136	173	39	102	163	152	176
76	13	1	168	159	22	125	143
120	70	14	150	145	144	145	143
58	86	119	98	100	101	133	113
30	65	157	75	78	165	145	124
11	170	91	4	110	17	133	110

Gradient Direction

Source: https://learnopencv.com/histogram-of-oriented-gradients/

Histogram of Oriented Gradients

- To provide a compact representation, an image is divided into 8 x 8 cells and a feature descriptor is used to characterize a patch of the image
- The gradient of this patch contains 2 values (magnitude and direction) each pixel for a total of 8 x 8 x 2 = 128 numbers
- Calculating a histogram over a patch makes this representation more robust to noise
- Arrow shows the direction of gradient, and its length shows the magnitude
- Direction of arrows points to the direction of change in intensity and the magnitude shows how big the difference is



2	3	4	4	3	4	2	2
5	11	17	13	7	9	3	4
11	21	23	27	22	17	4	6
23	99	165	135	85	32	26	2
91	155	133	136	144	152	57	28
98	196	76	38	26	60	170	51
165	60	60	27	77	85	43	136
71	13	34	23	108	27	48	110

Gradient Magnitude

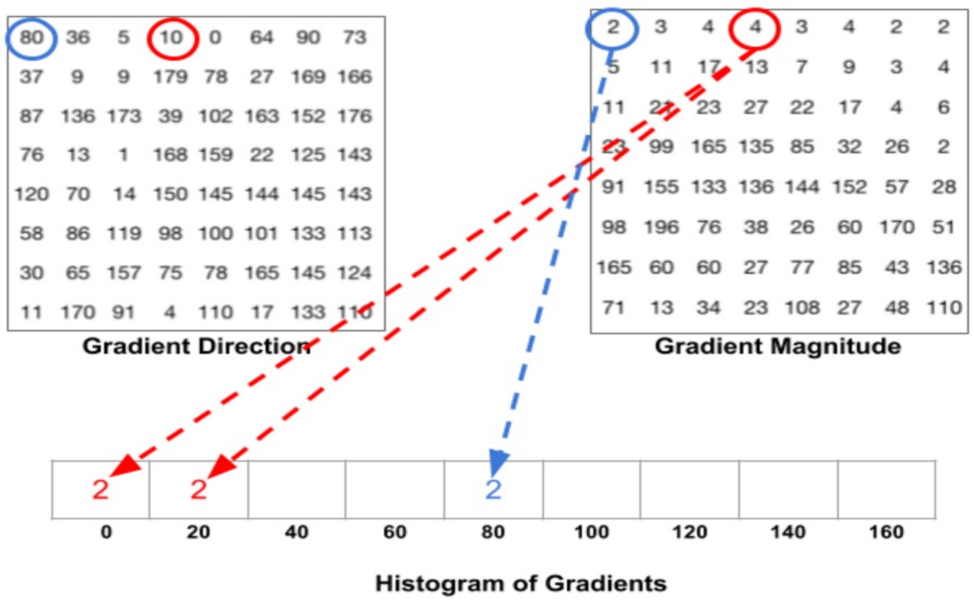
80	36	5	10	0	64	90	73
37	9	9	179	78	27	169	166
87	136	173	39	102	163	152	176
76	13	1	168	159	22	125	143
120	70	14	150	145	144	145	143
58	86	119	98	100	101	133	113
30	65	157	75	78	165	145	124
11	170	91	4	110	17	133	110

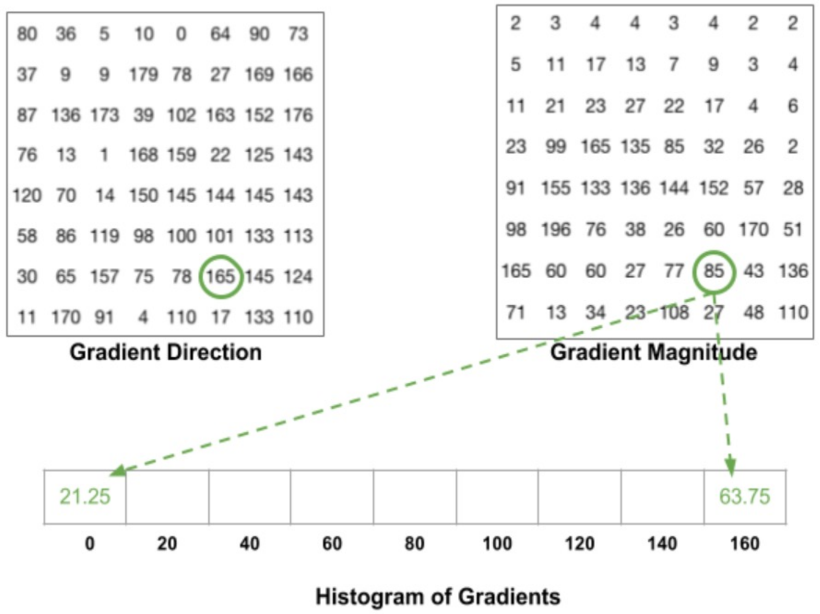
Gradient Direction

Source: https://learnopencv.com/histogram-of-oriented-gradients/

Histogram of Oriented Gradients

- The next step is to create a histogram of gradients in these 8×8 cells.
- The histogram is essentially a vector (or an array) of 9 bins (numbers) corresponding to angles 0, 20, 40, 60 … 160 (180^{θ}) [unsigned gradients]
- A bin is selected based on the direction, and the vote (the value that goes into the bin) is selected based on the magnitude

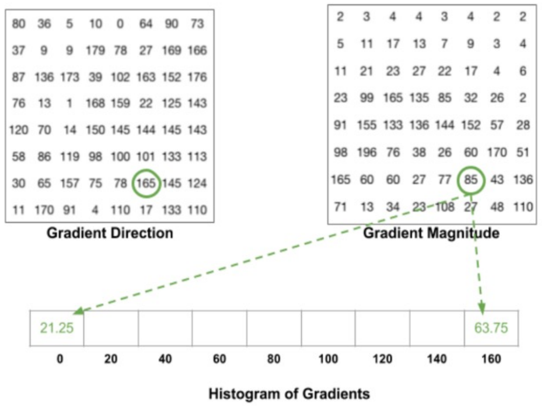


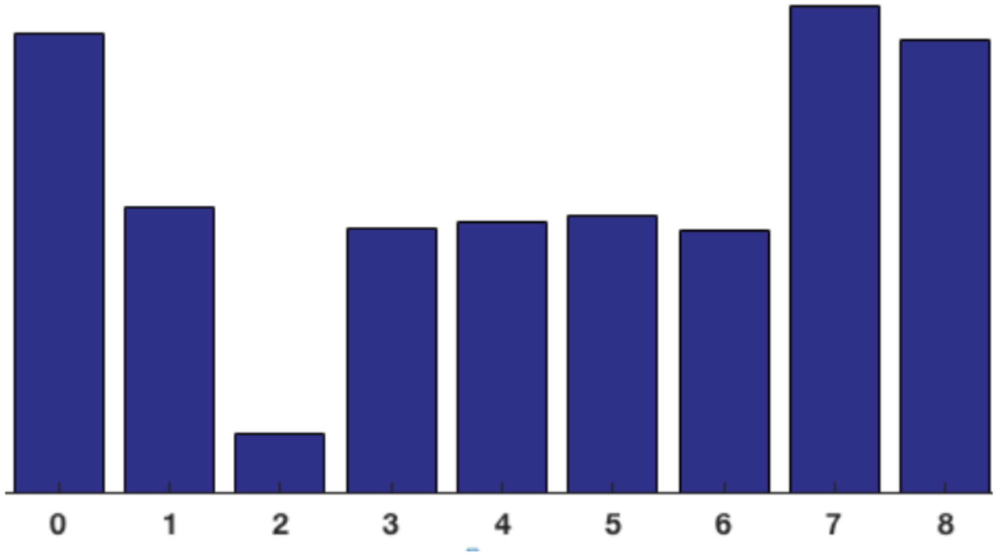


Source: https://learnopencv.com/histogram-of-oriented-gradients/

Histogram of Oriented Gradients

- The contributions of all the pixels in the 8×8 cells are added up to create the 9-bin histogram.





Source: https://learnopencv.com/histogram-of-oriented-gradients/

Histogram of Oriented Gradients

Step 4 : 16×16 Block Normalization

- A histogram is created based on the gradient of the image
- Gradients (magnitude) of an image are sensitive to overall lighting

$$[128, 64, 32]$$

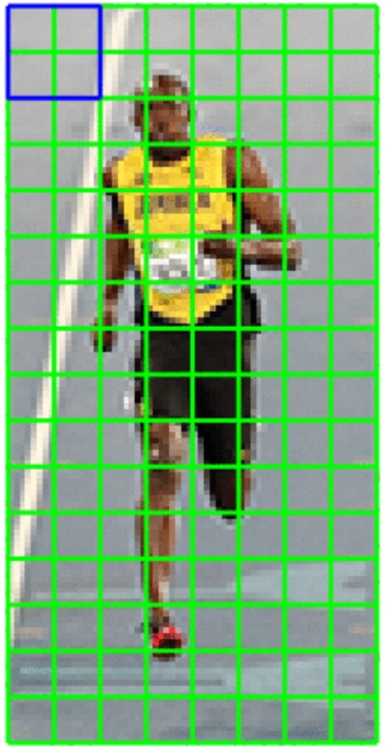
- L2 Norm of this vector is $\sqrt{128^2 + 64^2 + 32^2} = 146.64$
- Divide each element of this vector by 146.64: a normalized vector [0.87, 0.43, 0.22]

$$2 \times [128, 64, 32] = [256, 128, 64]$$

- normalizing [256, 128, 64] will result in [0.87, 0.43, 0.22], which is the same as the normalized version of the original

Histogram of Oriented Gradients

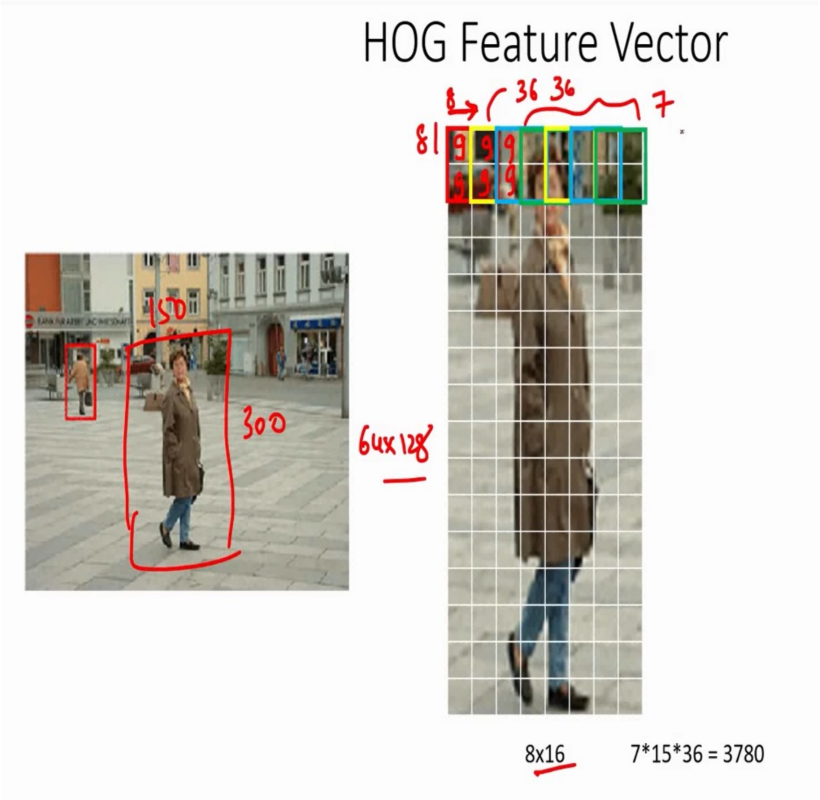
- A 16×16 block has 4 histograms which can be concatenated to form a 36 x 1 element vector
- It can be normalized the way a 3×1 vector is normalized.
- The window is then moved by 8 pixels and a normalized 36×1 vector is calculated over this window and the process is repeated.

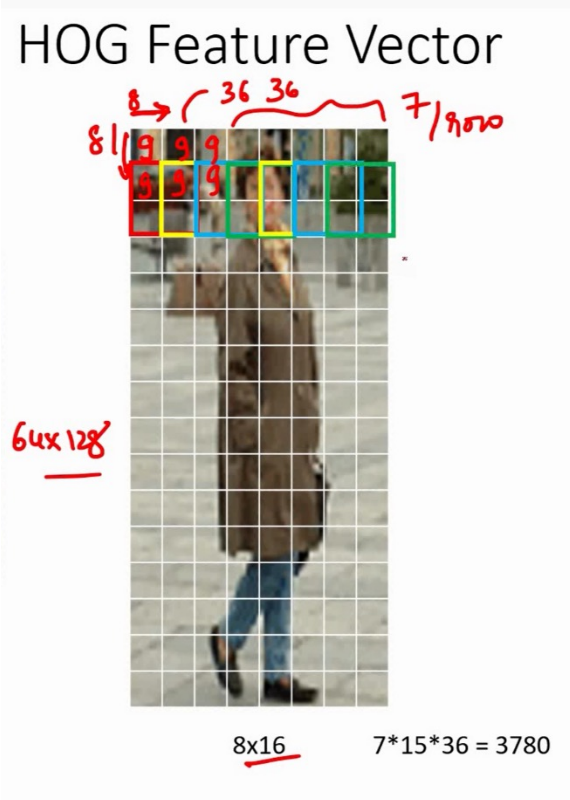


Source: https://learnopencv.com/histogram-of-oriented-gradients/

Histogram of Oriented Gradients

- This normalization is done to reduce the effect of changes in contrast between images of the same object





Source: https://www.youtube.com/watch?v=28xk5i1_7Zc
C34 | HOG Feature Vector Calculation | Computer Vision | Object Detection | EvODN

Histogram of Oriented Gradients

Step 5 : Calculate the Histogram of Oriented Gradients feature vector

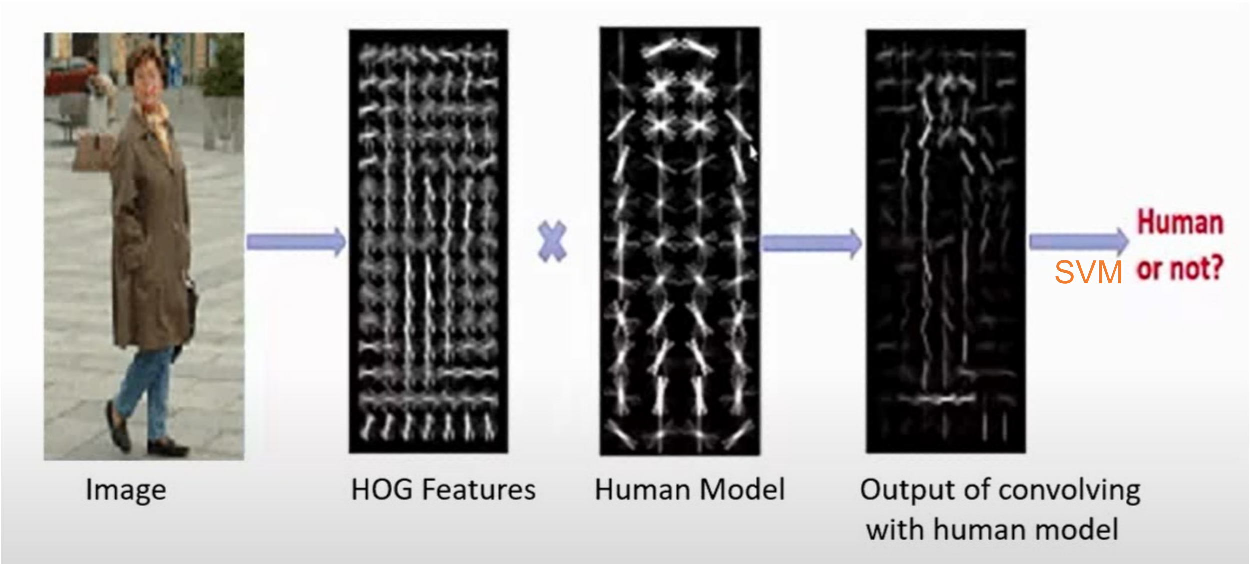
- To calculate the final feature vector for the entire image patch, the 36×1 vectors are concatenated into one vector

64 x 128

- There are 7 horizontal and 15 vertical positions making a total of 7 x 15 = 105 positions
- Each 16×16 block is represented by a 36×1 vector.
- So when we concatenate them all into one vector
- We obtain a 36×105 = 3780-dimensional vector

Histogram of Oriented Gradients

- Pedestrian detection using HOG



Source: Navneet Dalal and Bill Triggs: "Histograms of Oriented Gradients for Human Detection" CVPR-2005

Thank you for your attention