**اللهم علمنا ما ينفعنا، وانفعنا بما علمتنا، وزدنا علما "سُبْحَانَكَ لا عِلْمَ لَنَا إِلَّا مَا عَلَّمْتَنَا إِنَّكَ أَنْتَ الْعَلِيمُ الْحَكِيم"**

**CV-Image Processing**

Lecture 1

* Spatial Filtering technique: is used directly on pixels of an image. Mask is usually considered to be added in size so that it has specific center pixel. This mask is moved on the image such that the center of the mask traverses all image pixels.

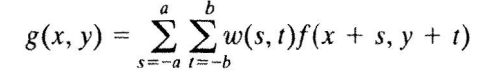
[ [Link](https://www.geeksforgeeks.org/spatial-filtering-and-its-types/) ]

* **Classification on the basis of Linearity: There are two types:**

1. Linear Spatial Filter 2. Non-linear Spatial Filter

* **The spatial filter consists of:**

1. Neighborhood, (typically a small rectangle).
2. Predefined operation that is performed on the image pixels by the neighborhood.

* **The linear spatial filter for image of M x N:**
* The spatial filter should be odd size
* The mask falls outside the edge:

1. Ignore the edges: The resultant image is smaller than the original
2. Pad with zeros: Introducing unwanted artifacts

* **Spatial Correlation and Convolution:** [ [Link](https://www.geeksforgeeks.org/difference-between-convolution-vs-correlation/) ]
* Correlation is the process of moving a filter mask over the image and computing the sum of products at each location.



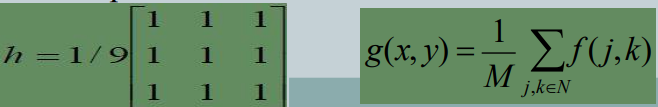
* The mechanics of convolution are the same except the filter first rotate by 180



* **Vector Representation of Linear Filtering:** = ***Z***
* **Smoothing Linear Filters:** (average, mean) are used for blurring and for noise reduction.
* A Low - Pass filter allows low spatial frequencies to pass unchanged, but suppress high frequencies. The low lass filter smoothies or blurs the image. Reduce noise and small details. **The elements of the mask must be positive**. **Sum of mask elements is 1.**

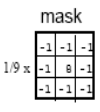
[ [Link](https://www.youtube.com/watch?v=07qT2L1ZKQA) ]

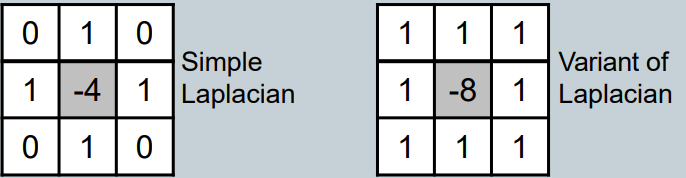
* Tends to reduce noise, but also obscures fine detail.
* A spatial averaging filters in which all coefficients are **equal** sometimes is called a **box filter**.
* A spatial averaging filters in which the coefficients are **different** sometimes is called a **Weighted Average**. [ [Link](https://www.youtube.com/watch?v=FvepLLicu0s) ]



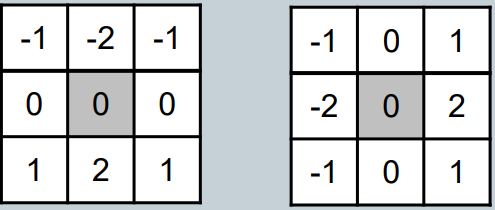
* **Order-Statistic (Nonlinear)Filters:** based on ranking
* It is based on the ordering the pixels contained in the image area encompassed by the filter. It replaces the value of the center pixel with the value determined by the ranking result. Edges are better preserved in this filtering.

1. Minimum filter: 0th percentile filter is the minimum filter. The value of the center is replaced by the smallest value in the window. **Reduce salt** [[Link](https://www.youtube.com/watch?v=wjZ5YtMkxDs)]
2. Maximum filter: 100th percentile filter is the maximum filter. The value of the center is replaced by the largest value in the window. **Reduce pepper** [ [Link](https://www.youtube.com/watch?v=HnwIqBWTsD0) ]
3. Median filter: Each pixel in the image is considered. First neighboring pixels are sorted and original values of the pixel is replaced by the median of the list. [[Link](https://www.youtube.com/watch?v=3kwH3Rb5LJ4)]

* Median filters are effective in the presence of impulse (**salt and pepper**) noise because of its appearance as white and black dots superimposed on an image.
* Sharpening Spatial Filters (derivative filter):
* Sharpening (i.e., high-pass filters): Highlight fine detail or enhance detail that has been blurred. The elements of the mask contain both **positive and negative** weights. **Sum of the mask weights is 0.**
* Useful for emphasizing transitions in image intensity (edges). Sharpening filters are based on spatial differentiation.
* Spatial Differentiation: Differentiation measures the rate of change of a function
* The formula for the 1st derivative:
* The formula for the 2nd derivative:
* The 2nd derivative is more useful for image enhancement than the 1st derivative. Stronger response to fine detail. Simpler implementation.
* Laplacian: One of the simplest sharpening filters

****

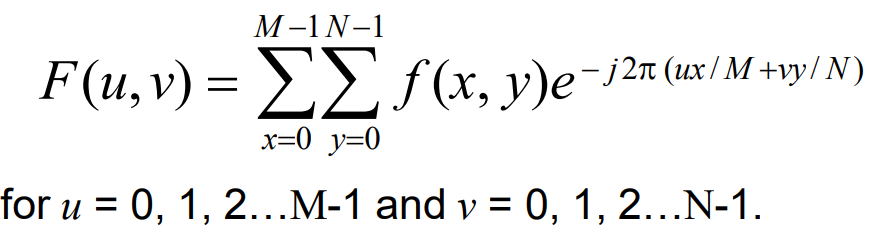
* Sobel filters: are typically used for edge detection. [ [Link](https://www.youtube.com/watch?v=Yz7h9L4gecQ) ]



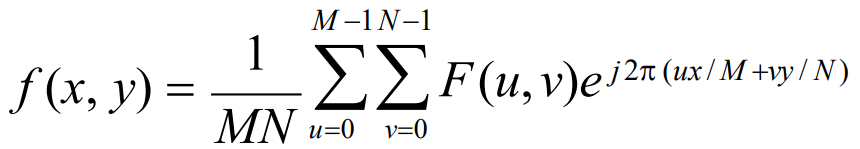
* Unsharp Masking: Sharpened image: Subtracting a blurred version of an image from the image itself
* **Sharpened image = original image – blurred image**

**Lecture 2**

* **The Discrete Fourier Transform (DFT):** transform image from special domain to frequency domain. [ [Link](https://www.geeksforgeeks.org/frequency-domain-filters-and-its-types/) ]



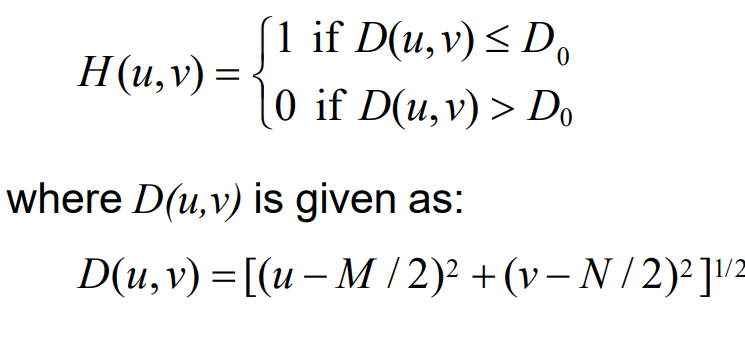
* Features from an image can even sometimes be seen in the Fourier spectrum of the image.
* **The Inverse DFT:**



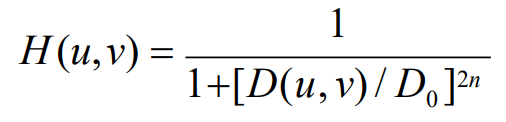
* **The DFT and Image Processing:**

1. Image pre-processing
2. Transform image from special domain to frequency domain
3. Apply filter
4. Inverse the transformation, transform image from frequency domain to special domain
5. Image post-processing

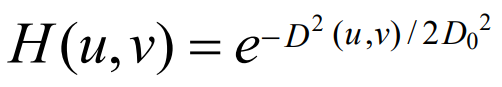
* **Smoothing Frequency Domain Filters**
* Smoothing is achieved in the frequency domain by dropping out the high frequency components.
* G(u,v) = H(u,v)F(u,v)
* Ideal Low Pass Filter: Simply cut off all high frequency components that are a specified distance **D0** from the origin of the transform. [ [Link](https://www.geeksforgeeks.org/matlab-ideal-lowpass-filter-in-image-processing/) ]
* The higher the value of D0, the greater the details in the image



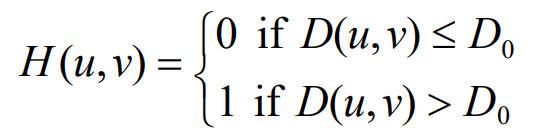
* Butterworth Low pass Filters: is used for image smoothing in the frequency domain. It removes high-frequency noise from a digital image and preserves low-frequency components. [ [Link](https://www.geeksforgeeks.org/matlab-butterworth-lowpass-filter-in-image-processing/) ]



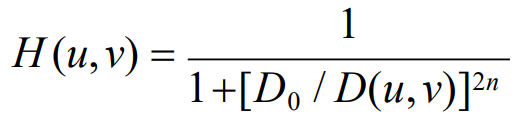
* Gaussian Low pass Filters: used for reducing noise (high frequency components) and blurring regions of an image. [ [Link](https://www.geeksforgeeks.org/apply-a-gauss-filter-to-an-image-with-python/) ]
* Mask is symmetric, calculated by Gaussian function.



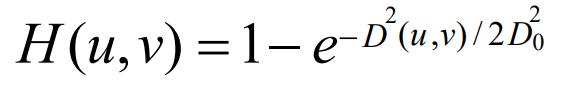
* **Sharpening Frequency Domain Filter**
* Ideal High Pass Filters: [ [Link](https://www.geeksforgeeks.org/matlab-ideal-highpass-filter-in-image-processing/) ]
* The higher the value of D0, the less sharpening.



* Butterworth High Pass Filters: [ [Link](https://www.geeksforgeeks.org/matlab-butterworth-highpass-filter-in-image-processing/) ]



* Gaussian High Pass Filters:



* Lecture 3
* Image restoration: attempts to restore images that have been degraded

1. Identify the degradation process and attempt to reverse it
2. Similar to image enhancement, but more objective

* The purpose of image restoration is to "compensate for" or "undo" defects which degrade an image.
* Degradation comes in many forms such as motion blur, noise, and camera misfocus.
* Degradation of images can have many causes:

1. Defects of optical lenses.
2. Nonlinearity of the electro-optical sensor.
3. Graininess of the film material.
4. Relative motion between an object and camera.
5. Wrong focus.
6. Atmospheric turbulence in remote sensing or astronomy, etc.
7. The objective of image restoration is to reconstruct the original image from its degraded version.

* The sources of noise in digital images arise during image acquisition (digitization) and transmission

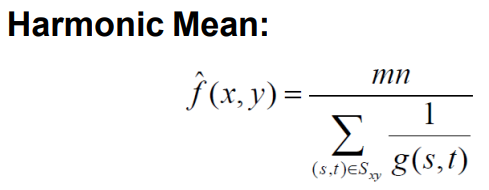
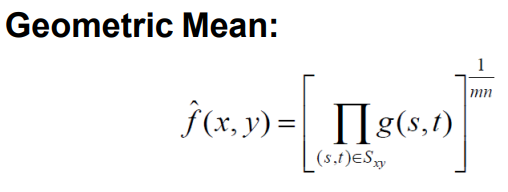
1. Imaging sensors can be affected by ambient conditions
2. Interference can be added to an image during transmission

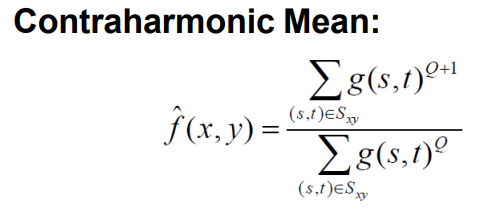
* Noise: g(x, y) = f (x, y) +η(x, y)
* There are many different models for the image noise term η(x, y):

1. Gaussian: Most common model
2. Rayleigh
3. Erlang
4. Exponential
5. Uniform
6. Impulse: Salt and pepper noise

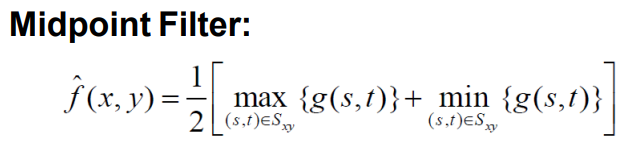
* Filtering to Remove Noise:

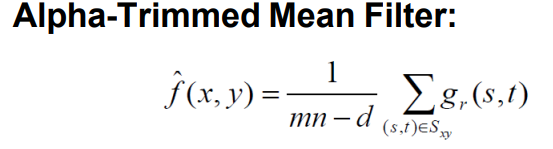
1. Arithmetic mean filter (average filter ): simple smoothing filter Blurs the image to remove noise
2. Geometric Mean: Achieves similar smoothing to the arithmetic mean, but tends to lose less image detail
3. Harmonic Mean: Works well for salt noise, but fails for pepper noise, and for other kinds of noise such as Gaussian noise

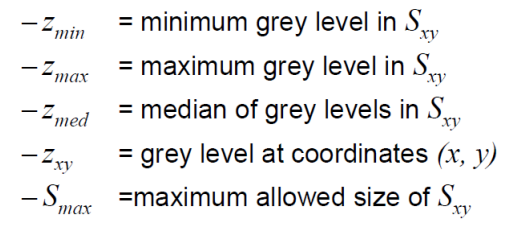
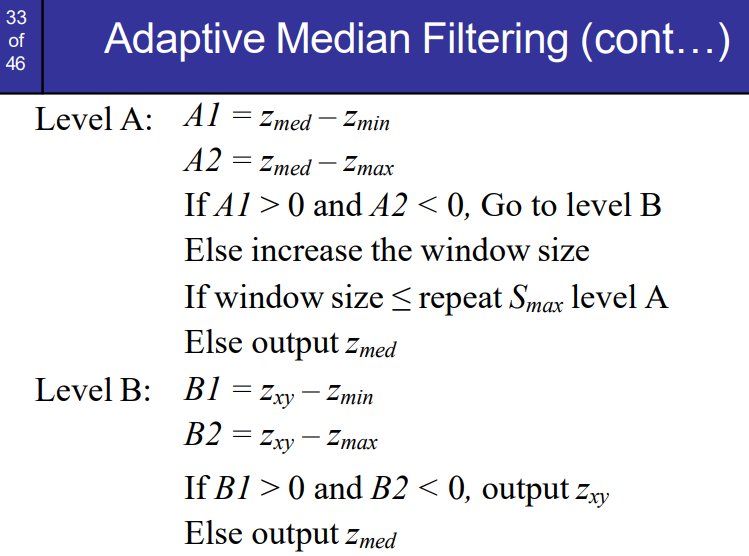


1. Contraharmonic Mean: Q is the order of the filter and adjusting its value changes the filter’s behavior
2.  Positive values of Q eliminate pepper noise
3. Negative values of Q eliminate salt noise

* Order Statistics Filters:

1. Median filter: Excellent at noise removal, without the smoothing effects that can occur with other smoothing filters, particularly good when salt and pepper noise is present.
2. Max and min filter: Max filter is good for pepper noise and min is good for salt noise.
3. Midpoint filter: Good for random Gaussian and uniform noise.
4. Alpha trimmed mean filter: delete the d/2 lowest and d/2 highest grey levels, gr(s, t) represents the remaining (m\*n – d) pixels

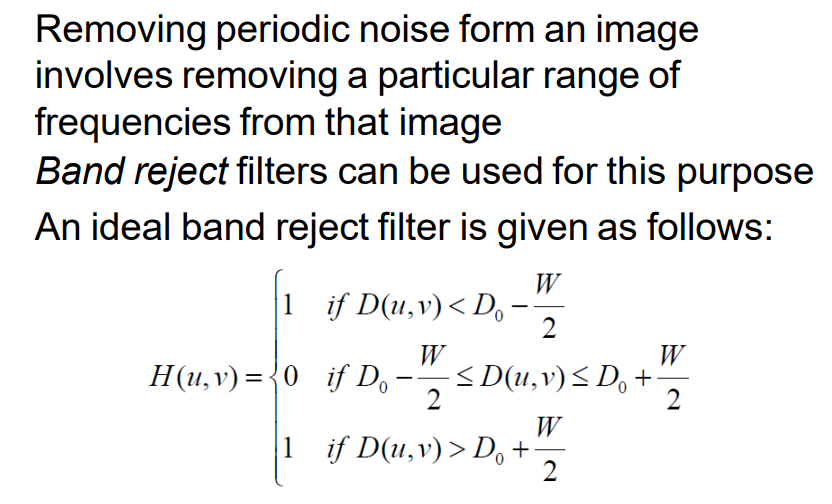
* Adaptive Filters:
* The behavior of adaptive filters changes depending on the characteristics of the image inside the filter region.
* Adaptive Median Filtering:
* The adaptive median filter can handle much more spatially dense impulse noise, and also performs some smoothing for non-impulse noise
* The adaptive median filter is that the filter size changes depending on the characteristics of the image



* Has three purposes:

1. Remove impulse noise
2. Provide smoothing of other noise
3. Reduce distortion

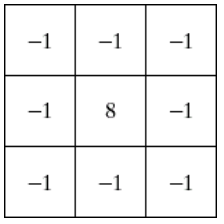
* Periodic Noise:
* Arises due to electrical or electromagnetic interference, frequency domain techniques in the Fourier domain are most effective at removing periodic noise
* Band Reject Filters: Removing periodic noise form an image involves removing a particular range of frequencies from that image (ideal band reject filter).



* **Lecture 4**
* **Segmentation**
* Segmentation attempts to partition the pixels of an image into groups that strongly correlate with the objects in an image.
* Algorithms are based on one of the 2 properties:

1. Discontinuity: partition an image based on the abrupt changes in intensity, such edges in an image.
2. Similarity: partitioning an image into regions that are similar according to a set of predefined criteria. Thresholding, region splitting and merging.

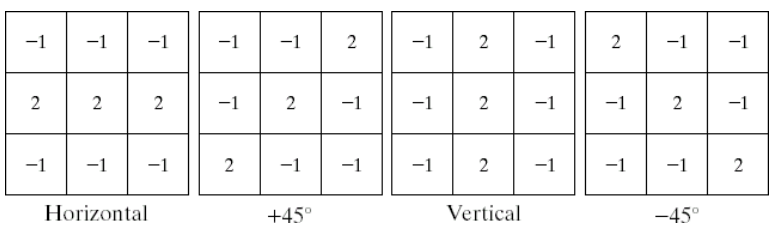
* **Detection Of Discontinuities: Points – Lines – Edges**

1. Point detection:

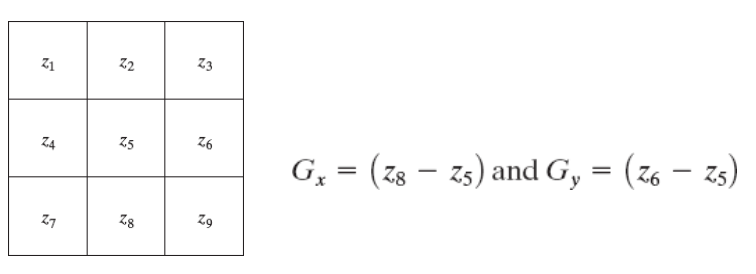
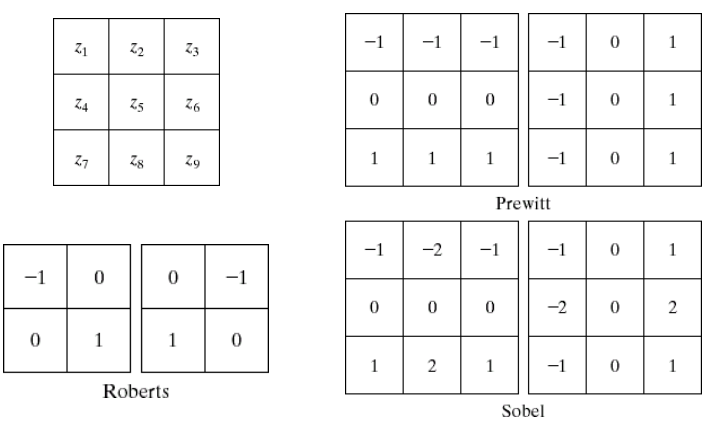
* Using Laplacian filter
* Point has been detected if |R| >= T (T non-negative threshold)
* The musk coefficients sum to zero. Mask response is zero in constant gray areas.

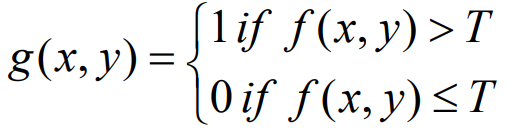
1. Line Detection:

* The masks below will extract lines that are one pixel thick and running in a particular direction.



1. Edge Detection:

* An edge is a set of connected pixels that lie on the boundary between two regions.
* The first step to object recognition.
* 1st derivative tells us where an edge is, 2nd derivative can be used to show edge direction.
* Derivative are extremely sensitive to noise.
* ∇f ≈ |Gx| + |Gy|
* **other filters used for edge detection:**
* **Edge Detection Problems:**
* there are is too much detail (Overcome: smooth image before edge detection)
* The Laplacian (2nd-order derivative) is typically not used by itself as it is too sensitive to noise.
* **Laplacian Of Gaussian:**
* The Laplacian of Gaussian (or Mexican hat) filter uses the Gaussian for noise removal and the Laplacian for edge detection.
* **Thresholding**
* Thresholding is usually the first step in any segmentation approach
* Single value threshold:



* **Basic Global Thresholding:**
* Based on the histogram of an image Partition the image histogram using a single global threshold
* Success depend on: how well the histogram can be partitioned
* **Problems With Single Value Thresholding:**
* Single value thresholding only works for **bimodal histograms**
* Images with other kinds of histograms need more than a single threshold (multilevel thresholding T1, T2)
* Effect of illumination on thresholding Illumination and Reflectance
* I(x, y) determined by illumination source, r(x, y) determined by characteristics of image objects
* **Algorithm:**

1. Select an initial estimate for T (typically the average grey level in the image)
2. Segment the image using T to produce two groups of pixels: G1 consisting of pixels with grey levels >T and G2 consisting pixels with grey levels ≤ T
3. Compute the average grey levels of pixels in G1 to give μ1 and G2 to give μ2
4. Compute a new threshold value: T = (μ1 + μ2) / 2.
5. Repeat steps 2 – 4 until the difference in T in successive iterations is less than a predefined limit

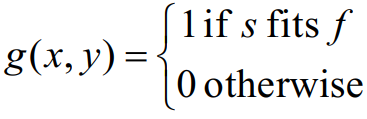
* The algorithm works very well for finding thresholds when the **histogram is suitable**
* **Basic Adaptive Thresholding:**
* To solve problems that basic global threshold cannot solve
* Works: divide an image into sub images and threshold these individually
* Subdivide the troublesome sub images for more success
* **Lecture 5**
* Morphological Processing:
* Morphological image processing (or morphology) describes a range of image processing techniques that deal with the shape (or morphology) of features in an image.
* Morphological operations can be used to remove imperfections in the segmented image and provide information on the form and structure of the image, after segmentation.
* Structuring Elements, Hits & Fits:

1. Fit: All pixels in the structuring element cover on pixels in the image
2. Hit: Any on pixel in the structuring element covers an on pixel in the image

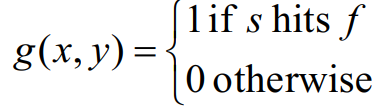
(If at least there is one pixel in image not match with the structure element)

* Structuring elements can be any size and make any shape.
* The structuring element is moved across every pixel in the original image to give a pixel in a new processed image.
* The value of this new pixel depends on the operation performed: erosion and dilation

1. Erosion:

* Erosion of image f by structuring element s is given by: **(f Θ s)**
* Erosion can split apart joined objects, strip away extrusions.
* Erosion shrinks objects.

1. Dilation

* Dilation of image f by structuring element s is given by **(f sꚚ)**
* Dilation can repair breaks, repair intrusions
* Dilation enlarges objects
* Compound Operations:

1. Opening

* The opening of image f by structuring element s, denoted **(f ○ s)** is simply an **erosion** followed by a **dilation**
* **f ○ s = (f Θ s) Ꚛs**

1. Closing

* The closing of image f by structuring element s, denoted **(f • s)** is simply a **dilation** followed by an **erosion**
* **f • s = (fꚚ s) Θ s**
* Morphological Algorithms:
* Boundary extraction
* Region filling
* Extraction of connected components
* Thinning/thickening
* Skeletonisation

1. Boundary Extraction:

* Extracting the boundary (or outline) of an object is often extremely useful
* Given by: **β(A) = A – (A Θ B)** (B structure element)

1. Region Filling:

* region filling attempts to fill that boundary with object pixels
* **Ꚛ B) ∩ , k= 1, 2, 3,…**
* Where is simply the starting point inside the boundary, **B** is a simple structuring element and is the complement of **A**
* This equation is applied repeatedly until is equal to
* **Lecture 6**
* **Covert Communications**

1. Cryptography
2. Information Hiding: Watermarking, Steganography

* Cryptography: Is the process of scrambling the secret message (plain-text) to hide its presence Cryptography
* Plaintext: any type of information in its original, unencrypted form.
* Cipher text: a message in its encrypted form.
* Encryption is the process of taking a plaintext message and converting it into cipher text, while decryption is the opposite of encryption which takes a cipher text message and converts it into plaintext.
* Watermark: is an imprint in a document file that you can use to prove authenticity of the file.
* **Steganography Terms:**

1. Cover image (Carrier): The original image into which the required secret message is embedded.
2. Payload (Message): is the secret massage that has to be embedded within the cover image.
3. Stego image (stego-object): is the final image obtained after embedded the payload into a given cover file.
4. Hiding Capacity: The size of information that can be hidden in the cover without degrading the quality of the cover image.

* **Fundamental Goals of Steganography:**
* Imperceptibility: is the perceptual difference between the cover and original image.
* Capacity: maximum amount of data that can be embedded into the cover image.
* Security: is referred as the resistance to the attacks.
* **Categories of Steganography:**

1. Substitution system techniques.
2. Transform domain techniques.
3. Spread spectrum techniques.
4. Statistical method techniques.
5. Distortion techniques.
6. Cover generation techniques.

* **Substitution System Techniques:** Replaces unneeded bits of a cover with the bits from the secret message. Use the **Least-Significant Bit (LSB)** method.
* 8 bit of message ----- 8 bytes of image
* Replace each bit of message with the least bit of image byte.
* **Lecture 7**
* **Image Classification**
* Challenges of classification:

1. Viewpoint Variation
2. Lighting Variation
3. Deformation
4. Background
5. Clutter Occlusion

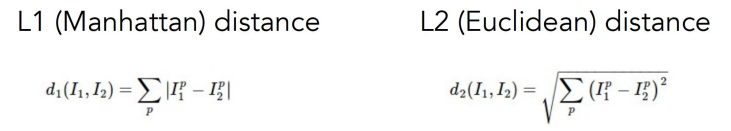
* **Images as High-Dimensional Vectors:** Divide space into different regions for different classes or define a distribution over space for each class.
* **Image Features and Dimensionality Reduction:**
* Feature selection: image from high dimensional to low dimensional

1. Relevant data
2. Irrelevant data

* **Training & Testing a Classifier**
* Training images > image feature
* Image feature & training labels > training ----------- learned classifier
* Test images > image feature > learned classifier > prediction
* **Nearest Neighbor (NN) Classifier: (KNN non-linear)**

1. Train: Remember all training images and their labels
2. Predict: Find the closest (most similar) training image, Predict its label as the true label

* **How to Define Distance Between Images:**



* **Hyperparameters in KNN:**
* K and distance: try them all and see what works best (for best accuracy)
* Dataset split to:

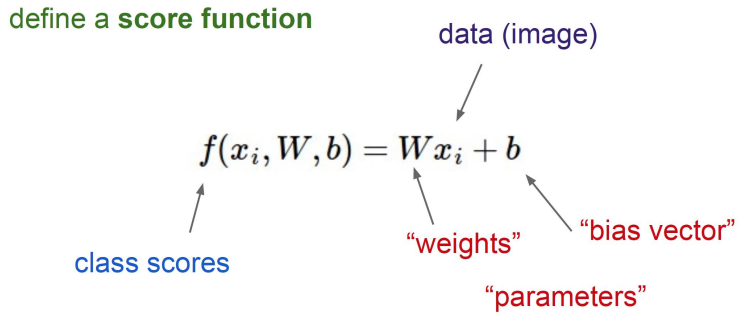
1. Fixed: train(for original model) , test(to understand generalizability) and validation data (validation set to choose the best hyberparameters)
2. Folds, try each fold as validation and average the result

* **kNN -- Complexity and Storage:**

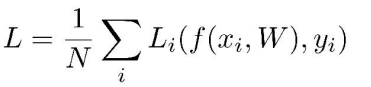
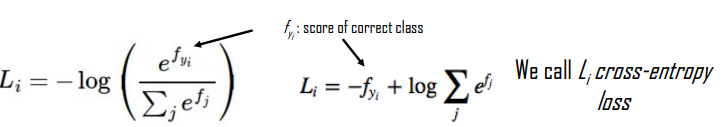
1. Train: O(1)
2. Test: O(MN)

* **Problems with KNN:**

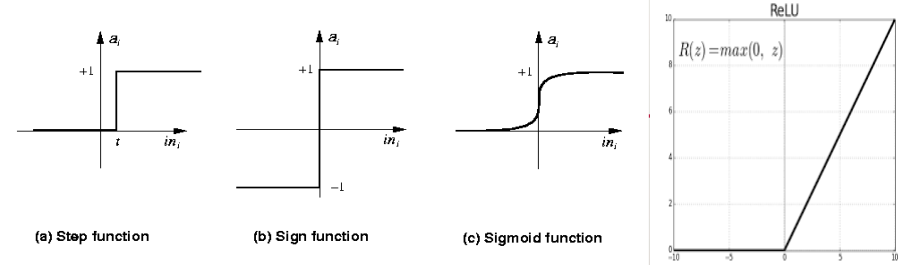
1. Distance Metrics
2. The Curse of Dimensionality: As the number of dimensions increases, the same amount of data becomes sparser (**متناثرة**, exponential).

* **Linear Classifier (SVN, Neural Network)**
* Store hyperplanes that best separate different classes
* We can compute continuous class score by calculating (signed) distance from hyperplane
* **Score function (hyper plane)**
* Xi image ------ M\*N\*3(RGB)\*Num\_of\_classes convert to flat (one column)
* Choose the best values of W, b using optimization algorithm
* Activation function takes the result of score function and convert it to probabilities
* Activation function like Sigmoid (binary classes), Soft max (multi classes)
* Best score function --- best hyper plane (best accuracy), be in the center of the margin(distance between two objects)
* **Interpretation:**

1. Geometric viewpoint (hyperplanes)
2. Template matching (visual viewpoint)
3. Algebraic viewpoint (f(x,w)=xW+b)

* **Loss function called (cost or objective):** define how good a classifier is based on the training data (unhappiness)
* Optimization algorithm choose the best values of W, b to achieve better accuracy and less lose (Linear), Choose K, L1, L2 in non-Linear
* Lose function = output (score) – Label
* **Cross-entropy loss:** is just one possible loss function
* SVM (max-margin) loss functions also used to be popular
* Over-fitting: classifier works well for train data but fails for test data.
* Under-fitting: classifier fails for train and test data.
* **Lecture 8**
* NEURAL NETWORKS
* **Class 1 Perceptron – Single Layer**
* Summation of each input each input neuron \* weight (It determines how much outputs will be affected by the inputs)
* Takes the output to activation function
* **Class 2 Multi-layer feedforward networks**
* Consist of:

1. Input layer: number of neurons
2. Hidden layers (Hyperparameter)
3. Feed-forward: Output links only connected to input links in the next layer.
4. Fully connected.

* **CLASS 3: RECURRENT NETWORKS:**
* No restrictions on connections. Behavior more difficult to predict/ understand.
* Apps: Voice to text.
* Output can return as input.
* ACTIVATION FUNCTIONS
* X (input)
* Step (x) = 1 if x >= t, else 0
* Sign(x) = +1 if x >= 0, else –1
* Sigmoid(x) = 1/(1+e^-x )
* Identity Function
* Relu(x) = max(0,X)
* What can perceptron represent?
* Functions which can be separated in this way are called Linearly Separable
* Only linearly Separable functions can be represented by a perceptron
* LEARNING

1. While epoch produces an error: Present network with next inputs from epoch, Presentation of the entire training set to the neural network.

**Err = T – O If Err <> 0 then Wj = Wj + LR \* Ij \* Err**

**End If, End While**

1. Training Value, T (expected output): When we are training a network we not only present it with the input but also with a value that we require the network to produce.
2. Error, Err: The error value is the amount by which the value output by the network differs from the training value. For example, if we required the network to output (T) **0** and it output (O) **1**, then **Err = -1**
3. Output from Neuron, O : The output value from the neuron
4. Ij : Inputs being presented to the neuron
5. Wj: Weight from input neuron (Ij ) to the output neuron
6. LR: The learning rate: This dictates how quickly the network converges. It is set by a matter of experimentation. **It is typically 0.1**

* LR determine by optimization algorithm called (Adam in CNN)
* IN ML: feature selection

1. Rapper method
2. Filter method
3. Embedded method