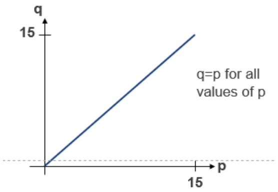
Vision Systems (Image) -> life before DL

- **Limitations of Computer Vision**
    - spatial information is "lost" when transforming to an image
    - Interpretation and understanding
    - Noise
    - Data intensive
    - contrast issues
- **Image Fundamentals**
    - Image function
        - f(x,y) consists of both spatial (x,y) and properties values f(x,y), g(x,y),etc..
        - *Image Sampling*:Digitization of the spatial coordinates *(x, y)*.(int)
        - Gray-level : Digitization of amplitude f(x,y). (2^G) where G is the number of bits in a byte which is used to represent the brightness of a pixel (0-255)
        - A continuous image *f(x, y)* can be represented by a *N* x *M* array shown below, where each element is a discrete quantity.
    - Image Coordinates
        - Pixel location
            - A pixel at location (n, m) has a numerical value which is the illumination (0-black)
        - Window Area
            - If a certain region of an image is of interest, we can encapsulate it with a window area for processing(ROI)
        - Image Properties
            - Spatial resolution
            - Effect of number of gray levels: from 8 to 1
- **Basic Image Processing**
    - Point by point (monadic) alteration of data on a global scale (g(i,j)=f[p(i,j)])
        - Involve the generation of new array by modifying pixel value at every single location based on a global rule
        - Process involves:
            - Obtaining pixel value of original array
            - Modifying it by a linear or non-linear operation
            - Placing the new pixel value in the corresponding location of the new array
        - Process is repeated and continued over entire array.
        - Identity Operator

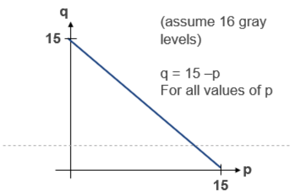
- 

  $$\begin{bmatrix} 12 & 15 & 0 \\ 0 & 2 & 14 \\ 11 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 12 & 15 & 0 \\ 0 & 2 & 14 \\ 11 & 2 & 3 \end{bmatrix}$$

  Input matrix (p) Output matrix (q)

  
  Input

  
  Output

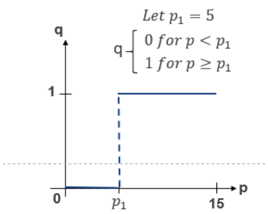
- Inverse Operator
  - 

    $$\begin{bmatrix} 12 & 15 & 0 \\ 0 & 2 & 14 \\ 11 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 3 & 0 & 15 \\ 15 & 13 & 1 \\ 4 & 13 & 12 \end{bmatrix}$$

    Input matrix (p) Output matrix (q)

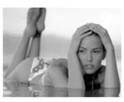
    
    Input

    
    Output

- Threshold Operator
  - 

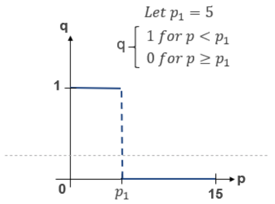
    $$\begin{bmatrix} 12 & 15 & 0 \\ 0 & 2 & 14 \\ 11 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

    Input matrix (p) Output matrix (q)

    
    Input

    
    Output

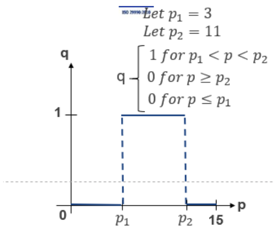
- Inverse Threshold Operator
  -



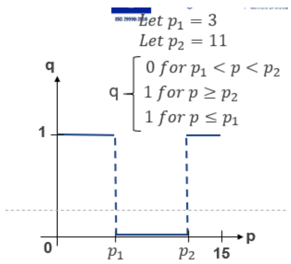
$$\begin{bmatrix} 12 & 15 & 0 \\ 0 & 2 & 14 \\ 11 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Input matrix (p) Output matrix (q)

Input Output

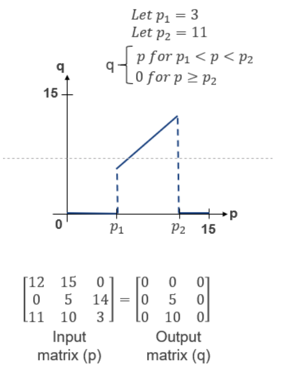
- Binary Threshold Interval Operator
  - 

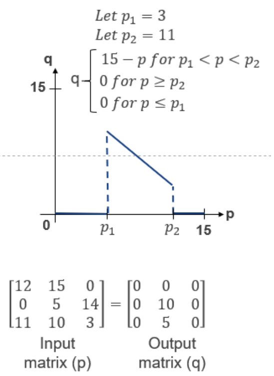
    $$\begin{bmatrix} 12 & 15 & 0 \\ 0 & 5 & 14 \\ 11 & 10 & 3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

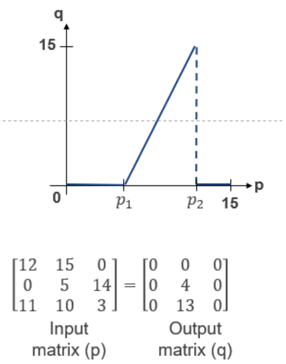
    Input matrix (p) Output matrix (q)
- Inverted Binary Threshold Interval Operator
  - 

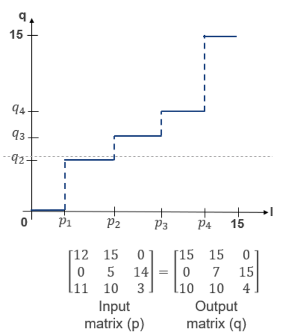
    $$\begin{bmatrix} 12 & 15 & 0 \\ 0 & 5 & 14 \\ 11 & 10 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

    Input matrix (p) Output matrix (q)
- Gray Scale Threshold Operator
  -



- Inverted Gray Scale Threshold Operator
    - 

- Stretch Operator
    - 

- Gray Level Reduction Operator
    - 

- Multiple points (dyadic) determination of elements of a new array of the image (*c(i, j) = f[a(i, j) , b(i, j)])*
    - The dyadic point-by-point operator uses the information contained at the same location in two images. The size of image does not change and can be linear or non-

linear.

- Image Addition
  - It can be used to reduce the effect of noise in the data. The value of output cij is given by over the range of values where k equals to number of samples.

    $$c_{ij} = \frac{(a_{ij} + b_{ij})}{k}$$

- Image Subtraction
  - It can be used to remove background or detect changes in a scene.

    $$c_{ij} = k(a_{ij} - b_{ij})$$

  - It is necessary to define the output as positive. rescaling or absolute of the output.

    Rescaling method:

    $$R_{ij} = (c_{ij} + 100)X\frac{255}{354} \qquad (7)$$

    is used to convert the values. R_{ij} will be 0 when c_{ij} is -100 (min value) and 255 when c_{ij} is 254 (max value).

    $$\underset{\textit{Input 1} = a_{ij}}{\begin{bmatrix} 0 & 12 & 142 & 255 \\ 1 & 6 & 40 & 254 \\ 24 & 0 & 20 & 255 \\ 30 & 2 & 10 & 240 \end{bmatrix}} \quad \underset{\textit{Input 2} = b_{ij}}{\begin{bmatrix} 14 & 11 & 9 & 253 \\ 3 & 5 & 39 & 254 \\ 11 & 1 & 19 & 255 \\ 18 & 2 & 11 & 256 \end{bmatrix}} \quad \underset{\textit{Output} = c_{ij}^*}{\begin{bmatrix} -14 & 1 & 133 & 2 \\ -2 & -19 & -60 & 254 \\ 15 & -1 & -60 & 254 \\ 0 & 0 & -100 & -15 \end{bmatrix}} \quad \underset{\textit{Output} = c_{ij}}{\begin{bmatrix} 82 & 74 & 158 & 74 \\ 71 & 59 & 29 & 255 \\ 83 & 72 & 29 & 255 \\ 72 & 72 & 0 & 62 \end{bmatrix}}$$

    Rescaling of the output method for image subtraction

- Convolution: Spatial Transformaiton
  - In a convolution, the pixel value is computed based on its neighbors. The new value does not replace the old matrix but is instead placed in the corresponding position in a new matrix. The location is then shifted by 1 and the process is repeated.

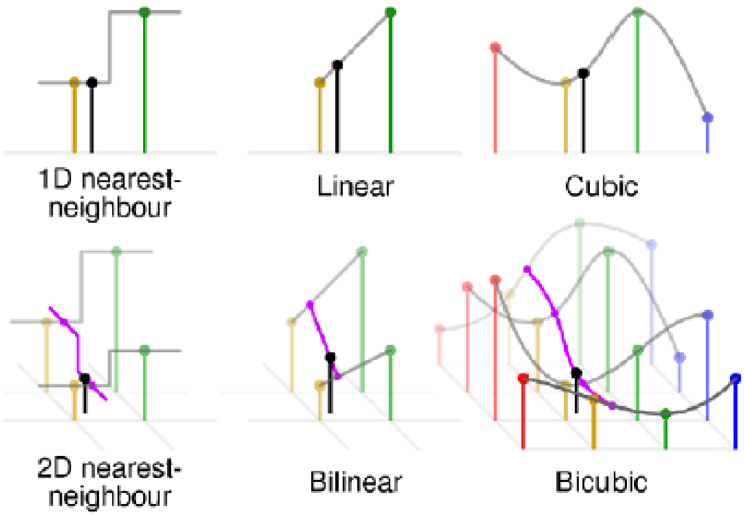
    $$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \times \begin{bmatrix} + & + & - \\ - & + & + \\ + & + & - \end{bmatrix} = \begin{bmatrix} \square & \square & \square \\ \square & e^* & \square \\ \square & \square & \square \end{bmatrix}$$

    $$e^* = +a + b - c - d + e + f + g + h - i$$

- Neighbors and Connectivity
  - Connectivity between pixels is an important concept used in establishing boundaries of objects and components of regions in an image.
  - 4-connectivity + 8-connectivity
- expand image border to avoid undesired effect
  - replicate
  - reflect

- wrap
- constant

- resize the image by interpolation



- the values of the output of a processing exceeds the usual range
  - process normalization

$$x_n = \frac{x - \text{oldMin}}{\text{oldMax} - \text{oldMin}}(\text{newMax} - \text{newMin}) + \text{newMin}$$

- **Binary Machine Vision**
  - Tend to have high values if they are part of an object of interest,Low values if they are not
  - Why to binary?
    - To identify objects of interest in the image. For example, we may wish to separate a visual target from its background.
    - For shape analysis, in which case the intensities of pixels are less significant than the shape of a region.
    - Enhancement of edges in an image. It is necessary to distinguish strong edges that correspond to object outlines (and features) from weak edges due to illumination changes,shadows, etc.
  - Thresholding operation
    - guideline:
      - Determine a threshold value
      - Pixels having gray value higher than threshold are given 1
      - Other lower than threshold are given 0
    - Main difficulties:
      - The distribution of the bright and dark pixels are usually not known
        - When the distribution of dark and bright pixels become more and more overlapped

- When there is substantial overlap, difficult to minimize classification error
- The only clue can be obtained from image histogram, but not always accurate.
  - If the gray levels of the object and the background are fairly close, the influence of noise may result in the object only appearing as a shoulder in the histogram. The threshold is also difficult to identify. This problem could be due to poor lighting conditions

- Global Thresholding
  - dividing the gray levels into bands and use thresholds to determine regions or to obtain object boundaries.
  - Process
    - After fixing T, we scan the given image f(x,y) of size N by N. A change in gray level from one band to the other denotes the presence of a boundary.
    - Scanning is done is two passes:
    - Pass 1: for each row in f(x,y), i.e. x = 0,1, …., N-1;
      - $$g1(x,y) = \begin{cases} LE & \text{If the level of } \boldsymbol{f(x,y)} \text{ and } \boldsymbol{f(x,y-1)} \text{ are in different bands of the gray scale} \\ LB & \text{If otherwise} \end{cases}$$

        where LE and LB are specified edge and background levels, respectively.
    - Pass 2: for each column in f(x,y), i.e., y =0, 1, ….., N-1.
      - $$g2(x,y) = \begin{cases} LE & \text{If the level of } \boldsymbol{f(x,y)} \text{ and } \boldsymbol{f(x-1,y)} \text{ are in different bands of the gray scale} \\ LB & \text{If otherwise} \end{cases}$$

        where LE and LB are specified edge and background levels, respectively.
    - Pass 1 and 2 are to detect changes in the y and x directions, respectively.

      $$g(x,y) = \begin{cases} LE & \text{If } \boldsymbol{g1(x,y) \text{ or } g2(x,y)} \text{ is equal to LE} \\ LB & \text{If otherwise} \end{cases}$$

      - The problem in this multi-bands situation is where to place the thresholds.
    - The combination of the results of the two passes will yield the boundary of the objects in the image.
- Component Labeling
  - Connected components labeling is a grouping operation that can change from pixel to region, which is a more complex unit.

- A region has a richer set of properties. It has shape and position properties as well as statistical properties of the gray levels of the pixels in the region.
- General Principles:
    - All the algorithms process a row of the image at a time
    - Modifications to process a sub-image rectangular window at a time are straight forward.
    - All the algorithms assign new labels to the first pixel of each component and attempt to propagate the label of a pixel to its neighbors.
- iterative algorithm (This algorithm selects the minimum label of its neighbors to assign to the pixel of interest.)
    - An initialization step where which pixel is given a unique label
    - A top-down pass, and on each row, a left-right pass in which the value of each non zero pixel is replaced by the minimum value of its non-zero neighbors in a recursive manner
    - A bottom-up pass, and a right-left pass with the same procedure in step 2.

	1	1		1	1	
	1	1		1	1	
	1	1	1	1	1	

(a)

	1	2		3	4	
	5	6		7	8	
	9	10	11	12	13	

(b)

	1	1		3	3	
	1	1		3	3	
	1	1	1	1	1	

(c)

	1	1		1	1	
	1	1		1	1	
	1	1	1	1	1	

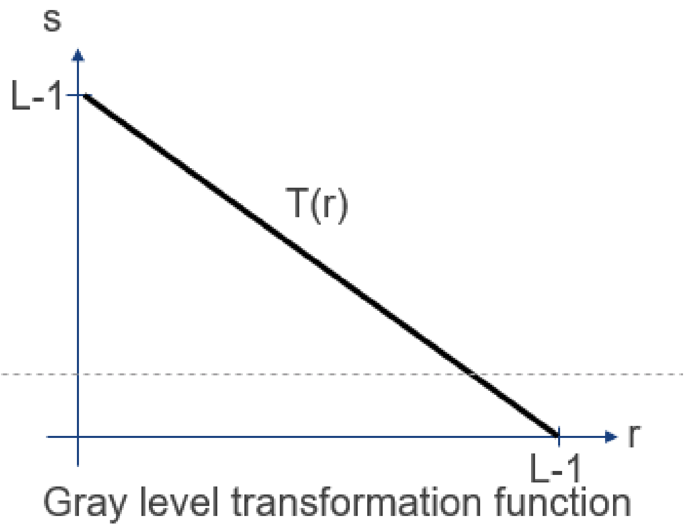
(d)

- **Image enhancement**
    - Frequency Domain methods : Based on modifying the Fourier transform of an image
        - We simply compute the Fourier transform of the image to be enhanced, multiply the result by a filter (rather than convolve in the spatial domain), and take the inverse transform to produce the enhanced image.
        - The idea of blurring an image by reducing its high frequency components, or sharpening an image by increasing the magnitude of its high frequency components is intuitively easy to understand. However, computationally, it is often more efficient to implement these operations as convolutions by small spatial filters in the spatial domain. Understanding frequency domain concepts is important, and leads to enhancement techniques that might not have been thought of by restricting attention to the spatial domain.
        - filter
            - Low pass filtering involves the elimination of the high frequency components in the image. It results in blurring of the image (and thus a reduction in sharp transitions associated with noise). An ideal low pass filter would retain all the low frequency components, and eliminate all the high frequency components. However, ideal filters suffer from two problems: *blurring* and *ringing*. These problems are caused by the shape of the associated spatial domain filter, which

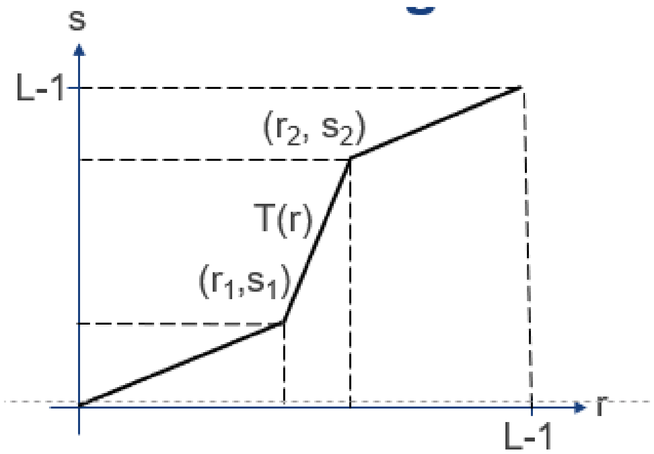
has a large number of undulations. Smoother transitions in the frequency domain filter, such as the Butterworth filter, achieve much better results.

- 

- Spatial Domain methods: Direct manipulation of the pixels
    - Point Processing
        - g(x,y)=T[f(x,y)] T: operations f:input
        - 3*3 Mask
        - Image Negative (displaying medical images and photography)

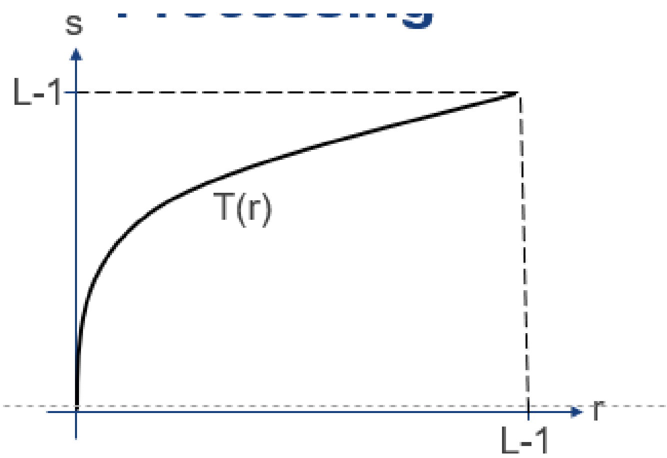
            

            
        - Contrast stretching
            -



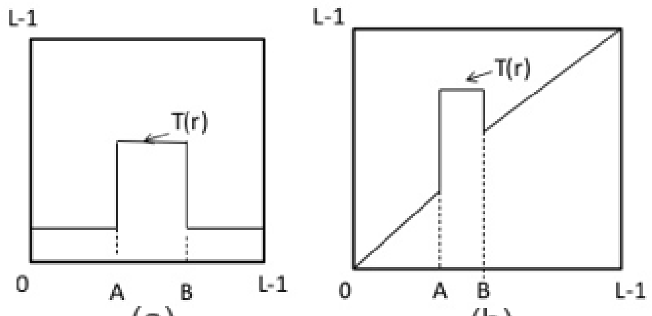
Original Low Contrast | Contrast Stretching | Binary thresholding

- Low contrast results from:
    - Poor illumination
    - Lack of dynamic range
    - Wrong aperture setting during acquisition
- Compression of Dynamic Range
    -

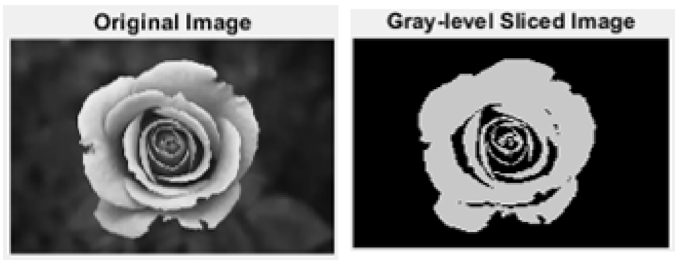


Original large dynamic range | After transformation

- **s=c.log(1+|r|)**
- where (1+|r|) yield 0 to 6.4.
- To scale to 0 to 255 (8 bit system), set c = 255/6.4,where c is a scaling constant.

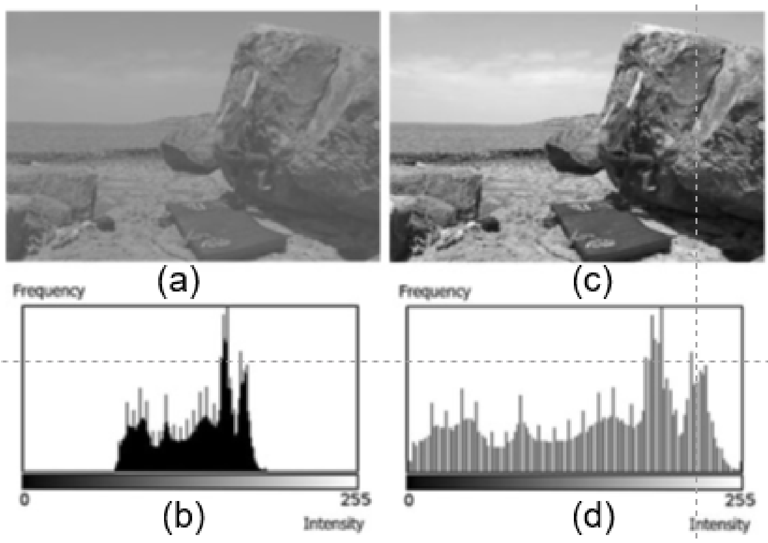
- Gray Level Slicing
  - Used to highlight specific range of gray levels ofinterest.
  - T(r) in (a) produces a binary image. T(r) in (b)brightens the gray levels ranging from A to B, but preserves the background andtonalities in the image.
  - (c) shows an image and (d) shows the resulting image after the image has been processed by the T(r).
  - 

(a) (b)



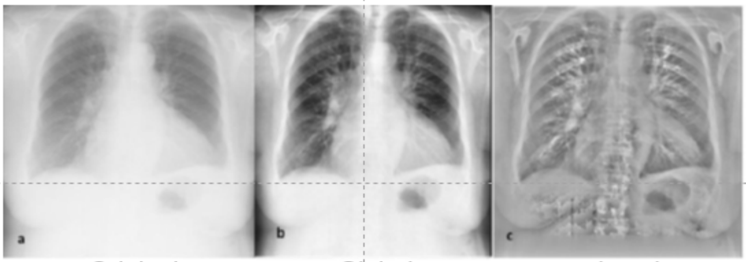
(c) (d)

- Histogram processing
  - Problem with histogram:
    - provides only statistical information
    - Zero information about the spatial distribution of the pixel values
    - Possible solutions: divide image into regions and do further analysis
  - Histogram Equalization -> increase image contrast



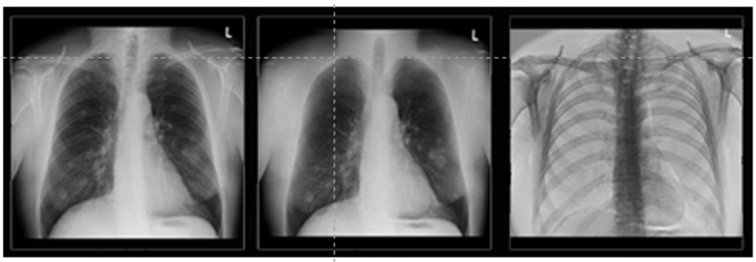
- Contrast limited adaptive histogram equalization (CLAHE)

- Local Enhancement -> enhance details over small areas



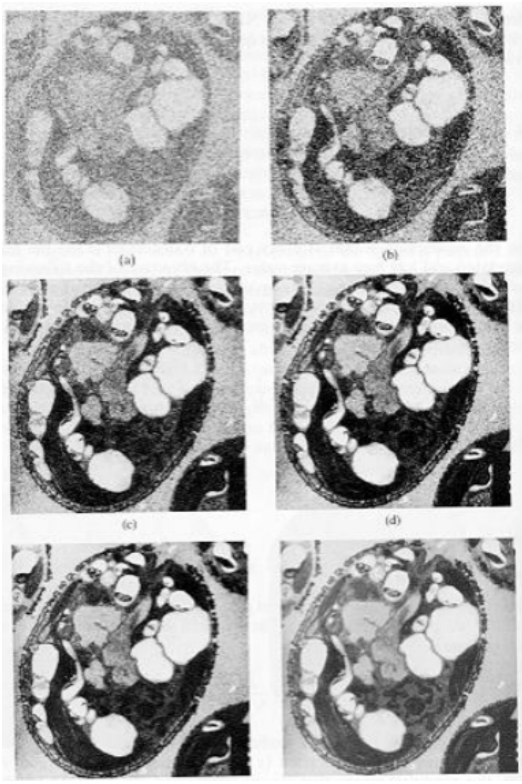
Original image | Global enhancement | Local enhancement using mask

- Image Subtraction -> g(x,y)=f(x,y)-h(x,y)

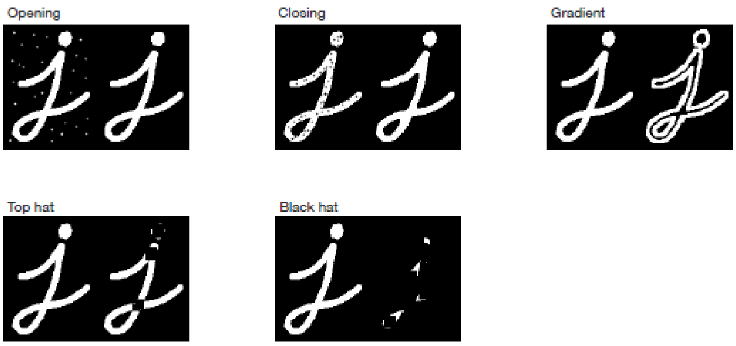


Standard (High kV) | Soft Tissue (Low kV) | Bone (Subtracted)

- Image Averaging (noise reduction by averaging (a) a typical noisy image; (b)-(f), results of averaging 2, 8, 16, 32, and 128 noisy images.)



- Morphological operation



  - Shrinking, growing
    - Shrinking is performed by removing a layer of border pixels -> erosion
    - Growing is performed by attaching a layer of pixels -> dilation
    - Morphological operation involves kernel / structuring element; the size and the shape of kernel affects the rate and the style of shrinking and growing
  - closing=1. dilation 2.erosion
  - opening= 1. erosion 2. dilation
  - Top hat composite morphological operation is defined as the difference between input image and its opening
  - Black hat composite morphological operation is defined as the difference between input image and its closing
  - Gradient composite morphological operation is defined as the difference between the dilation and the erosion of a given image
- Spatial Filtering (base on masks)

- R=w1z1 + w2z2+…+w9z9 -> The center of the mask is placed at (x,y),the gray level of the pixel is then replaced by R.
- low-pass filters
  - Eliminate high frequency components in the Fourier Domain while leaving low frequencies un-touched.
  - High-frequency components characterize edges and other sharp details in an image. Hence,low-ass filters will cause <u>image blurring</u>.
  - Low-pass filter has a limitation in which it blurs out sharp details while removing noise.
  - Gaussian filtering -> uniform low pass filter
    - The kernel size should be around 6 x sigma (3 sigma for each side)
    - Commonly used to remove noise and detail
- High-pass filters(sharpening filter)

  $$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & 8 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

  3 x 3 high pass spatial filter

  - Note that the <u>sum of coefficients is equal to zero</u>. When the mask shown is over an area of constant or slow varying gray level, the output is zero or very small.
  - Eliminate low frequencies. These filters has the effect of sharpening edges and other details.
  - To highlight fine details in an image or to enhance detail that has been blurred
  - improvement -> high boost filtering
    - High boost = (A)(Original)-Low-pass = (A-1)(Original)+Original – Low-pass=(A-1)(Original) + High-pass
  - Derivative Filters

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

(a)

1	0		0	1
0	-1		-1	0

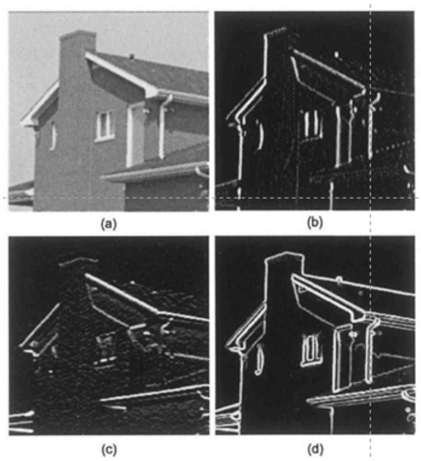
(b) Roberts

-1	-1	-1	-1	0	1
0	0	0	-1	0	1
1	1	1	-1	0	1

(c) Prewitt

-1	-2	-1	-1	0	1
0	0	0	-2	0	2
1	2	1	-1	0	1

(d) Sobel



On the left shows 2D edge enhancement using the Sobel filters.

(a) Original house image

(b) Gx image (vertical edges)

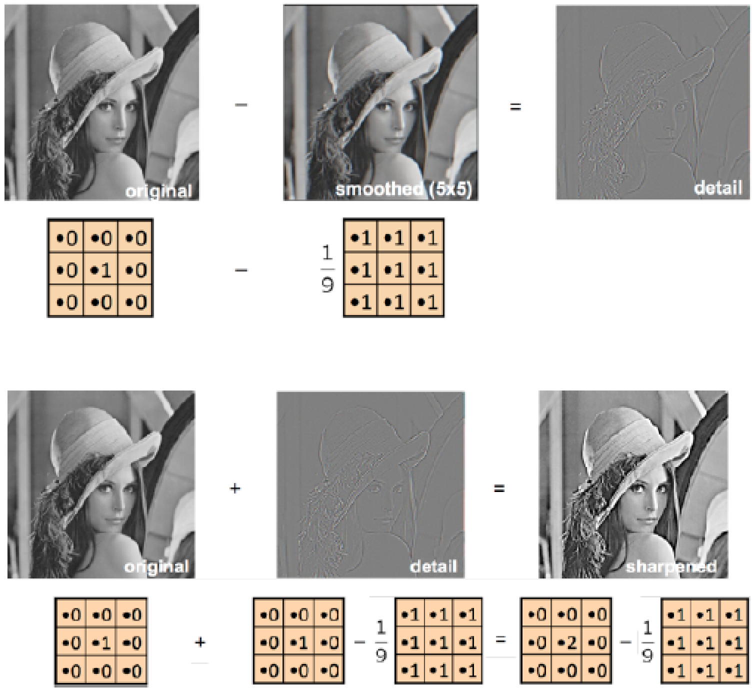
(c) Gy image (horizontal edges)

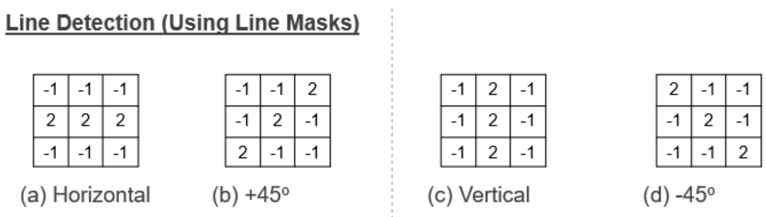
(d) Combined edges

- Smoothing Filters(neighborhood averaging\ mean filter)

  $$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

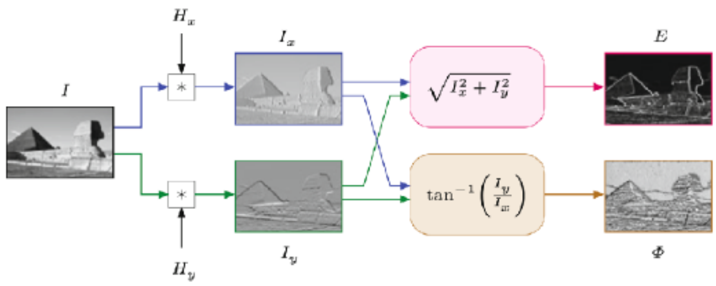
  - Removal of small details prior to large object extraction
  - Bridging gaps in lines or curves
  - Noise reduction
  - Median filter -> better than mean: it considers the neighbourhood around a pixel,order the values, determine the median value and output the median value
- Examples of non-linear filters are
  - median filtering for noise reduction, yet maintain the sharp details;
    - Effective in removing noise patterns consisting of strong and spike-like components while preserving sharp edges.
  - max filter: R= max {zk |k = 1, 2,...., 9} to find the brightest spot.
  - min filter, R= min {zk |k = 1, 2,...., 9} to find the dimmest spot.
- combinations



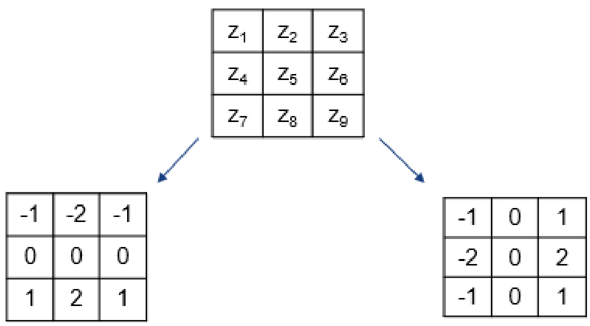
- **Image Segmentation**
  - base on discontinuity and similarity
  - **Discontinuity**
    - Segment an image based on abrupt changes in gray level.
    - The areas of interest are the detection of isolated points, lines and edges in an image.
    - using masks
      - 

      - Let Ra, Rb, Rc and Rd denotethe responses of the masks (a – d), from left to right.
      - All masks are run through an image. If at certainpoint in the image, |Ri|>|Rj|, for all i≠j, that pointis said to be more likely to associate with a line in the direction of mask i.
  - **Similarity**
    - Based on thresholding, region growing, and region merging and splitting.
  - **<u>Edge Detection</u>**
    - An edge is the boundary between two regions with relatively distinct gray level properties.
    - Assumption: regions are sufficiently homogenous so that transition between two regions can be determined on basis of gray-level discontinuity alone.

- Gradient Operators (value +direction)
  - definition



    - The gradient vector points in the direction of maximum rate of change.
    - In edge detection, an important quantityis the magnitude of ∇f⃗,
    - Gradient is commonly approximated by the absolute values: ∇f≈|G_x |+|G_y |
    - Direction of ∇f⃗, at(x, y) : α(x,y)=tan^(-1)(G_y/G_x)
  - Sobel Operator
    - providing both a differencing and a smoothing effect



Compute Gx at center point of 3 x 3 region

compute G_y at center point of 3 x 3 region

  - Canny edge detection
    - 1. Noise reduction: remove noise with 5 x 5 Gaussian filter
    - 2. Calculate intensity gradient:Filter image with Sobel kernel in both horizontal and vertical direction. In image processing, Sobel kernel is often used to calculate the range of change in pixels along x-direction and y direction
    - 3. Non-maximum suppression:check if an edge point is a local maximum in its neighbourhood in the direction of gradient
    - 4. Hysteresis thresholding: everything above maxVal is edge; every thing below minVal is not edge; For those in between, see connections
- Segmented in HSI color space:
  - A thresholding function based on color information in H and S Components. We rarely use I component for color image segmentation.
- Segmentation in RGB vector space:
  - A thresholding function based on distance in a color vector space.

- methods:
  - Region growing
  - Watershed
  - k-means clustering
  - Level-set
  - Fast marching
  - Mumford-Shah model
  - Random walker
  - Markov random fields
  - Simulated annealing
  - Expectation maximization
  - Active contour model
    - Snake

      ▪ As the contour moves, it tries to minimize the energy

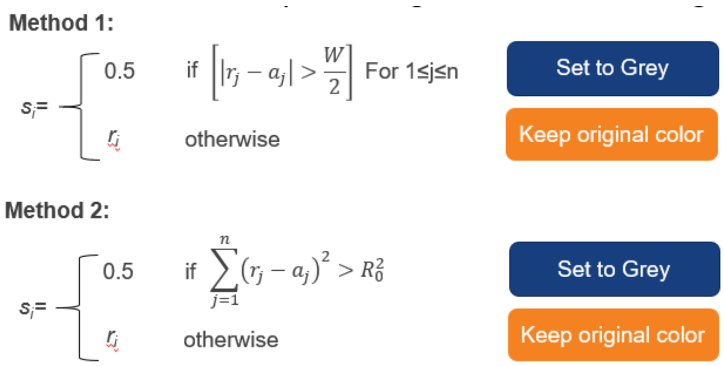
      $$E = \int_0^1 \left[\frac{1}{2} \left(\underbrace{\alpha \left| \mathbf{x}'(s) \right|^2}_{\text{the energy derived from the length / tension of the contour}} + \underbrace{\beta \left| \mathbf{x}''(s) \right|^2}_{\text{the energy derived from the smoothness of the contour}} \right) + \underbrace{E_{ext} \left(\mathbf{x}(s) \right)}_{\text{the energy derived from the image of interest}} \right] ds$$

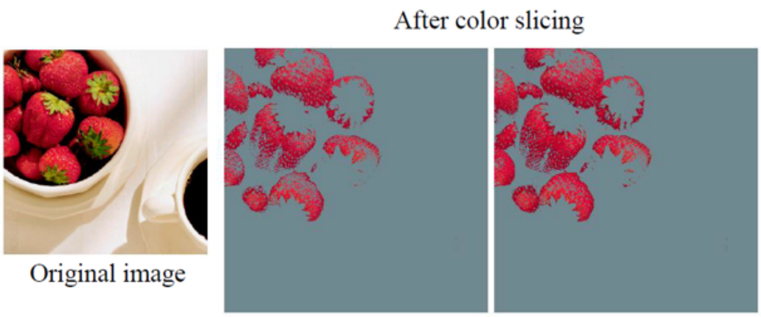
    - geometric active contour

      $$\frac{\partial u}{\partial t} = \underbrace{g(I)|\nabla u| \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right)}_{\text{smoothing force}} + \underbrace{g(I)|\nabla u|\nu}_{\text{balloon force}} + \underbrace{\nabla g(I) \nabla u}_{\text{image attraction force}}$$

      - Balloon force: decides if the curve is going to inflate or deflate
      - v is the parameter to play with
    - Active contour without edges
      - morphological_chan_vese
  - Active shape model
  - Active appearance model
- **Color Image Processing**
  - Hue:dominant color corresponding to a dominant wavelength of mixture light wave
    - specifycolor tone (redness, greenness, etc.)
    - depend on peak wavelength
  - Saturation:Relative purity or amount of white light mixed with a hue (inversely proportional to amount of white light added)
    - describe how pure the color is

    - dependon the spread (bandwidth) of light spectrum
    - reflect how much white light is added
  - Brightness:Intensity
  - image process
    - Pseudo-color image process: Assigning colors to gray values based on a specific criterion.
      - In some case there is no “color”concept for a gray scale image but we can assign “false” colors to an image. (Human can distinguish different colors better than different shades of gray)
    - Full color image process: Manipulation of real color images such as color photographs.
      - Per-color-componentprocessing: process each component separately.
      - Vector processing: treat each pixel as a vector to be processed.
      - color slicing

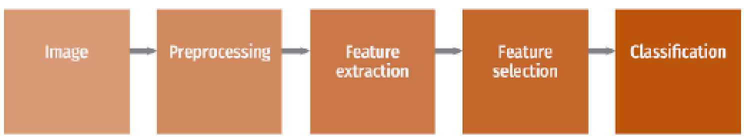




a b

FIGURE 6.34 Color slicing transformations that detect (a) reds within an RGB cube of width $W = 0.2549$ centered at (0.6863, 0.1608, 0.1922), and (b) reds within an RGB sphere of radius 0.1765 centered at the same point. Pixels outside the cube and sphere were replaced by color (0.5, 0.5, 0.5).

- Image classification

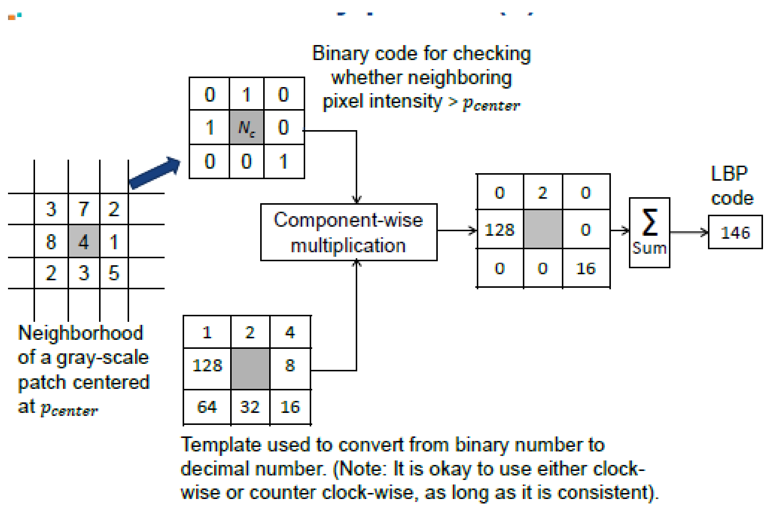


- preprocessing
  - Image adjustment (rotating,resizing and etc…)
  - Noise removal / image enhancement
  - Contrast adjustment (histogram equalization, CLAHE, subtraction from mean and etc…)
  - Perform transformation (Fourier, shearlet, wavelet, radon and etc …)
- Feature extraction

Category	Representative features
Color	Spectral peaks and histogram
Geometrical	Edges, lines, line widths, line relationships (e.g., parallel, perpendicular), circles, shapes, size of enclosed area
Statistical	Number of lines, area and perimeter, mean, variance, kurtosis, skewness, entropy
Time domain	Motion characteristics, speed, acceleration, trajectory
Frequency domain	Fourier coefficients and other time-frequency domains (such as discrete Cosine transformation, Gabor, Wavelet)

  - Feature: a piece of information useful for discrimination between classes
  - Some are visually related: Shapes, edges, orientations of lines and etc.
  - Many are not, such as histogram,statistical information and etc.
  - Texture: patterns or variations among pixels
  - LBP: Local binary pattern
    - use label“1”if the center pixel's value is smaller than the neighbour; otherwise, uselabel “0”.
    - LBP code

      The LBP code (the decimal value) for this pixel p_{center} is $LBP = \sum_{i=0}^{7} d_i \cdot 2^i$.



    - 1.Divide the image into cells(e.g.,16×16pixels).Build one histogram per cell,then average all histograms to form a final histogram for the entire image.
    - 2.For each pixel in a cell,compare it with its neighbours(various configurations are shown below).Where the neighbour pixel’s value is greater than that of the

center,score it as 1;otherwise score it as 0.Walking round all the neighbours then gives an array of binary numbers,and further convert it into decimal number.

- 3.For each cell,compute the histogram of the frequency of each decimal number. Optionally normalise the histogram.
- 4.Finally,average all the histograms to give a descriptor for the entire image.

- HOG: Histogram of oriented gradients
  - INFO

Clarification between Edge and Gradient		
Edge detection	Binary (yes/no)	Decision of edge detection task
Gradient	Continuous measurement	Used as features for other tasks

Gradient magnitude

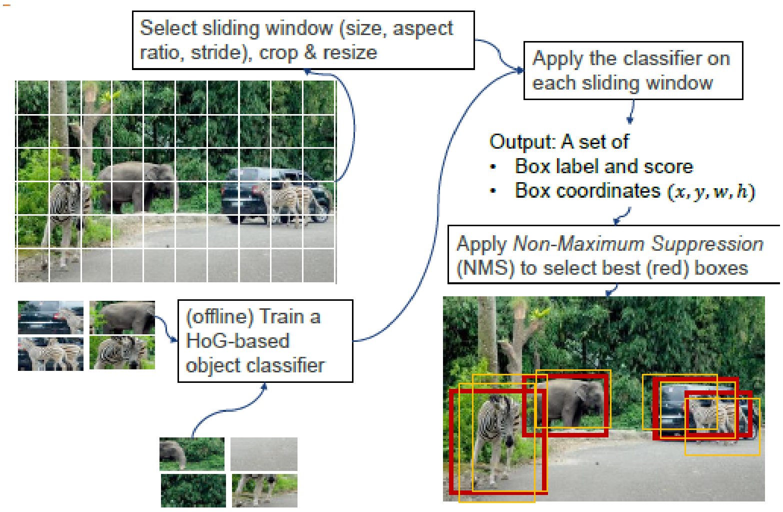
$$m(x,y) = \sqrt{\underbrace{(I(x+1,y) - I(x-1,y))^2}_{\text{gradient in } x \text{ direction using filter } [-1,0,1]} + \underbrace{(I(x,y+1) - I(x,y-1))^2}_{\text{gradient in } y \text{ direction using filter } [-1,0,1]^T}}$$

Gradient direction

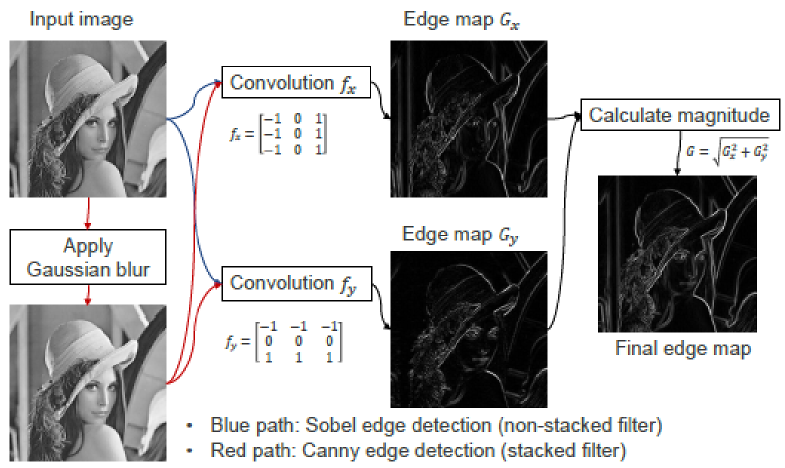
$$\theta(x,y) = \tan^{-1}\left(\underbrace{(I(x,y+1) - I(x,y-1))}_{\text{gradient in } y \text{ direction using filter } [-1,0,1]^T} / \underbrace{(I(x+1,y) - I(x-1,y))}_{\text{gradient in } x \text{ direction using filter } [-1,0,1]}\right)$$

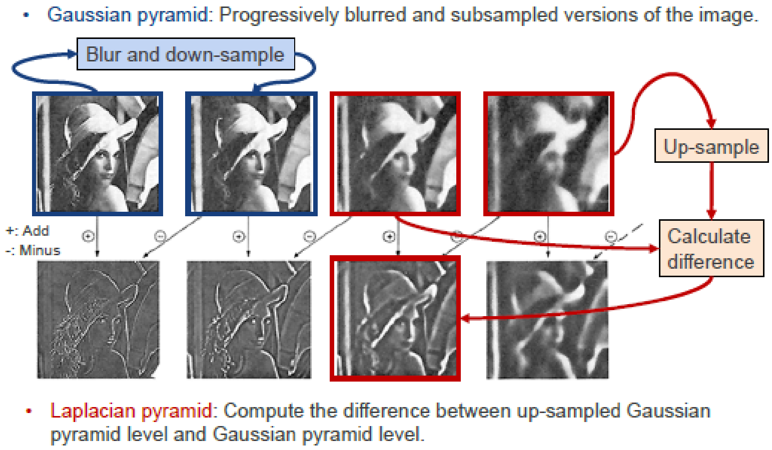
$I(x,y)$ = the image intensity at the pixel position (x,y)

	Dimension	Remark
Image	64×128	A fixed-size input image
Block	16×16	With 50% overlap, Total 7×15=105 blocks
Cell	8x8	Each block should consist of 2×2 cells
Feature	9-bin	Each cell has a HoG feature of 9-bin histogram
Total features	3780	105 (block) × 4 (cell) × 9 (bin) = 3780



- Combinations: CNN motivation: Stacked filters





- pre-trained CNN like on imagenet can be freeze and fine-tune(without fc)

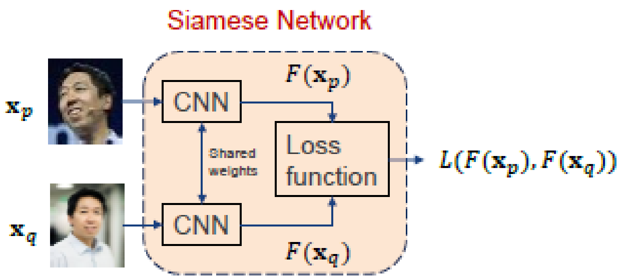
- Feature selection
  - Too many features generated; slow or not possible to train and classify
  - Some features are redundant; some are irrelevant
  - improve generalization by reducing overfitting
  - Some of the methods commonly used: Linear discriminant analysis\Genetic algorithm\Principal component analysis\Partical swarm optimization\Ants colony
- Classification
  - With all the features read/prepared,fed into classifier
  - Get algorithms to learn about the patterns under data set, rather than specify rules to determine class
  - Some of the classifiers commonly used: k nearest neighbour\Neural network\Decision tree\Naive Bayes
- Feature similarity evaluation
  - compare two vector loss function

- Euclidian distance: $f(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|^2 = (\mathbf{x}-\mathbf{y})^T(\mathbf{x}-\mathbf{y})$
- Cosine similarity distance: $f(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}\cdot\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n}(x_i^2)}\sqrt{\sum_{i=1}^{n}(y_i^2)}}$ Dot product

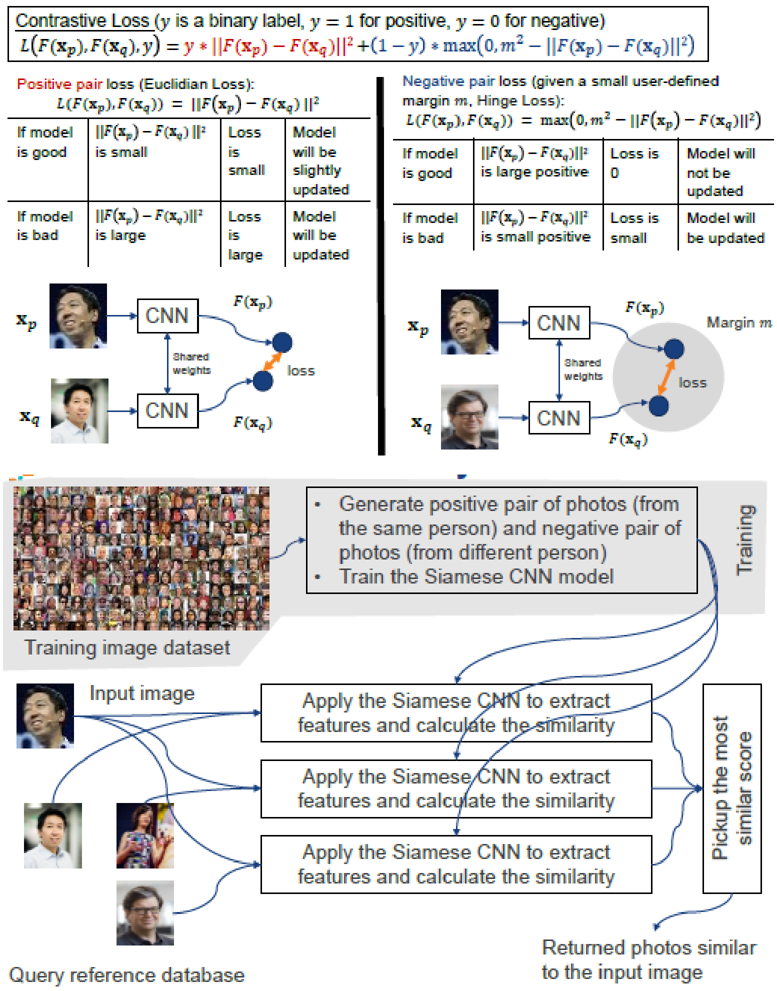
Euclidean norm (i.e., length of vector)

Scenario	Similar	Unrelated	Opposite
Input two vectors $\mathbf{x},\mathbf{y}$	Same direction	Nearly orthogonal	Opposite direction
Angle between them θ	Near 0 degree	Near 90 degree	Near 180 degree
Similarity score	Near 1	Near 0	Near -1

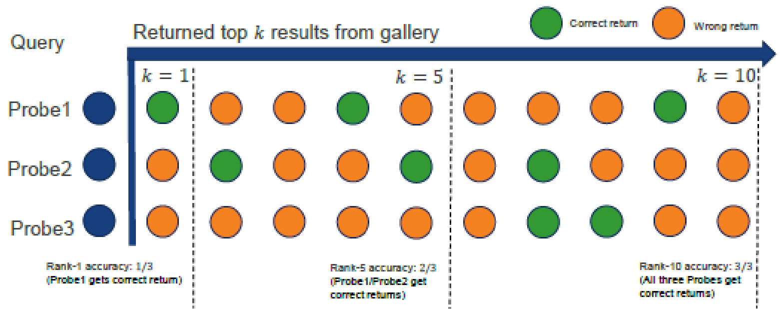
- Siamese network



  - Sample positive pairs$(\mathbf{x}p,\mathbf{x}q)$, with$(p,q)$of same class.
  - Sample negative pairs$(\mathbf{x}p,\mathbf{x}q)$, with$(p,q)$of different classes.
  - Forward pass using both inputs through the two networks (sharable weights). Back propagate through the two networks (the weights are updated with the sum of the two gradients).
  - Loss and framework

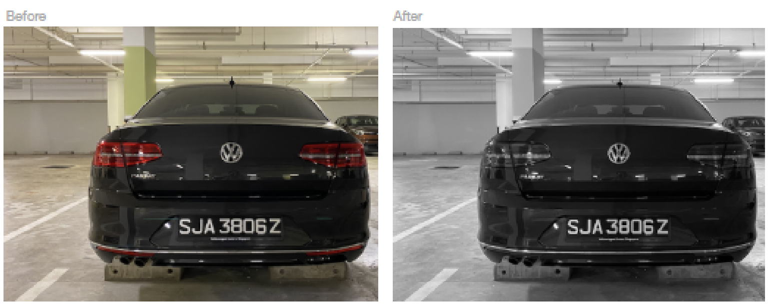


- Performance evaluation



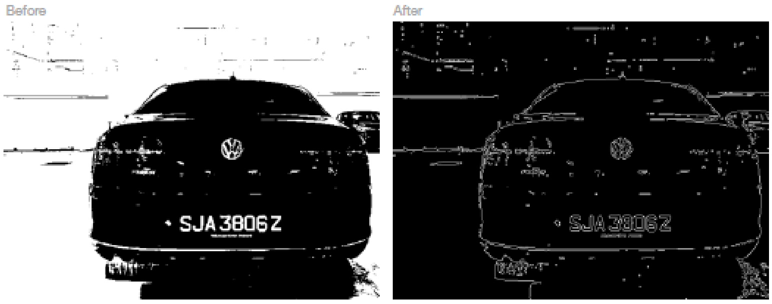
- Query: Given probes (e.g., 3 images) of the target person, evaluate each image in the dataset (gallery), rank the returned results in terms of the similarity scores.
- Rank-k accuracy: Whether the system gets correctly (at least one) in the top k returned results.
- mAP (*mean average precision*): Average precision for all probes.
  - Probe1: $AP = (1/1 + 2/4 + 3/9)/3 = 0.61$
  - Probe2: $AP = (1/2 + 2/5 + 3/7)/3 = 0.44$
  - Probe3: $AP = (1/7 + 2/8)/2 = 0.20$
  - Overall: $mAP = (0.61 + 0.44 + 0.2)/3 = 0.42$

- Main challenges
  - 1. View point variation
  - 2. illumination
  - 3. Occlusion
  - 4. Scale
  - 5.Deformation
  - 6. Background clutter
- Example: locate carplate in an image
  - RGB2GRAY

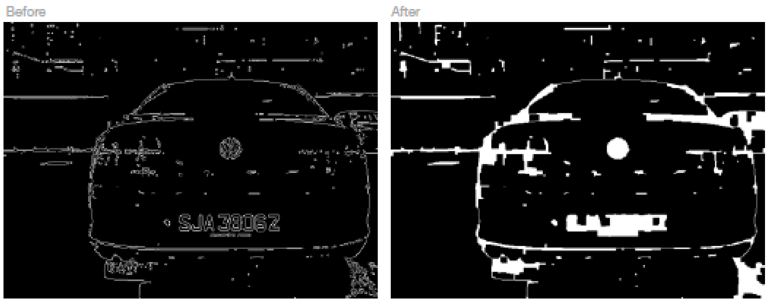


  - Otsu thresholding

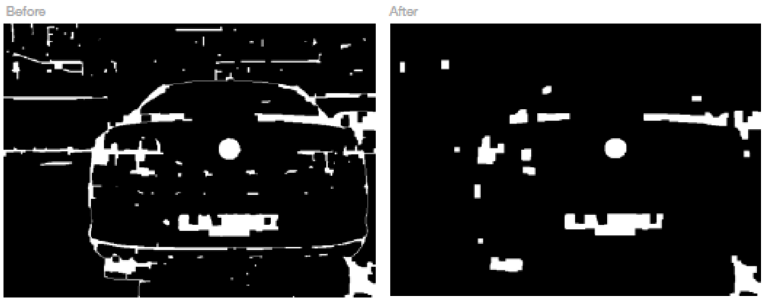
  - Canny:extract the salient features in the image, this is done through edge detection



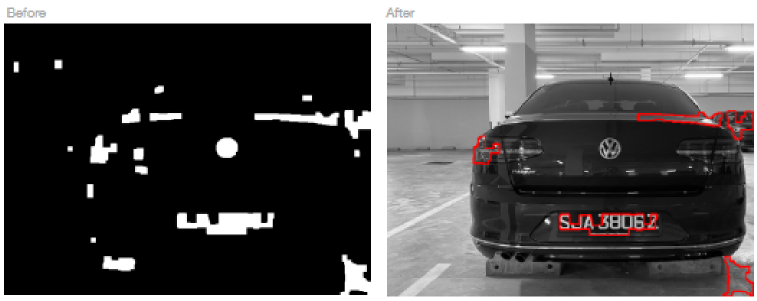
- dilate and erode the borders of the characters



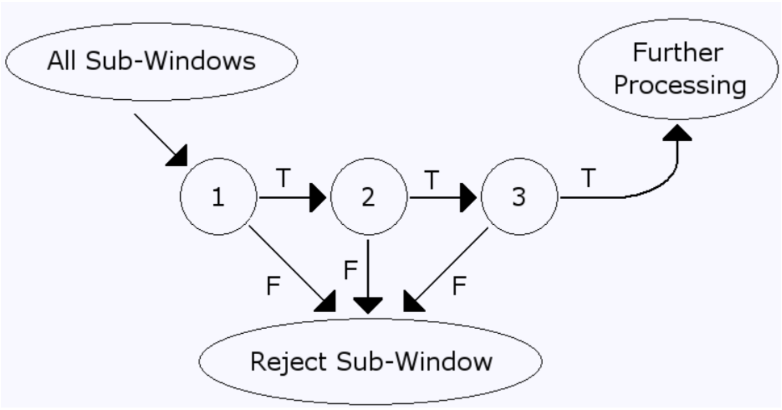
- To remove background noise, we apply erode followed by dilate 2 times



- Get some of of the largest contours in the background-remove image, and plot them out for inspection,Only the first 5 largest-area contours are highlighted and kept in the analysis



- Example: face detection
  - Viola-Jones algorithm -> real-time
    - AdaBoost

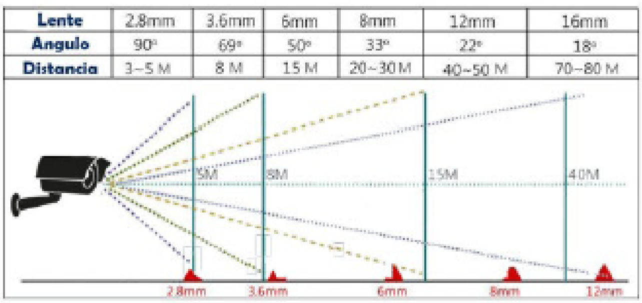


- pros and cons
  - Extremely fast computation, good for real-time detection
  - Position and scale invariant
  - Can be applied to other type of object detections
  - Good and effective only on frontal face
  - Not able to cope with faces that rotate around 45 degree
  - Sensitive to lighting conditions
- cascadeclassifier

- Key considerations of vision systems
  - Analytics objective

	Low-level	Middle-level	High-level
Input	Image	Image	Image/attributes
Output	Image	Attributes	Understanding
Examples	• Noise removal • Image sharpening	• Object detection and segmentation	• Scene understanding • Event recognition

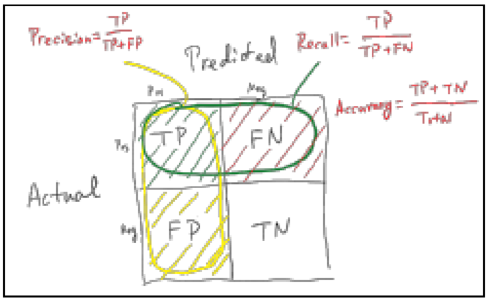
  - Acquisition
    - Light-Sensitive Camera: Measures from the visible part of the electromagnetic spectrum, e.g., RGB camera.
    - Multi/Hyper-Spectral Sensors: Measure from a broader part of the electromagnetic spectrum (than the light-sensitive cameras), e.g., infrared, thermal.
    - Range Sensor: A device that measures the distance from the camera to a target, e.g., laser, radar, Lidar.
    - Additional equipment needed (such as external lighting)



Lente	2.8mm	3.6mm	6mm	8mm	12mm	16mm
Angulo	90°	69°	50°	33°	22°	18°
Distancia	3~5 M	8 M	15 M	20~30 M	40~50 M	70~80 M

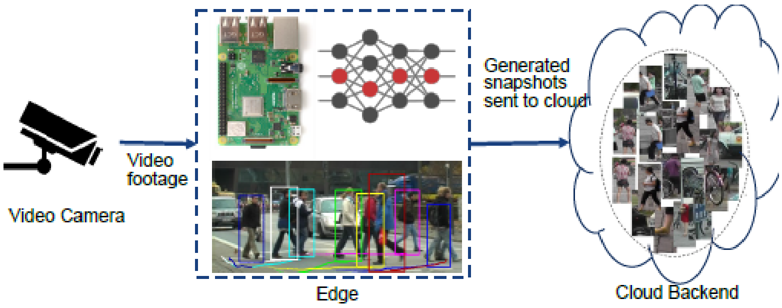
  - Performance index

- What detection rate and false alarm are acceptable?
- Test scripts, red teaming

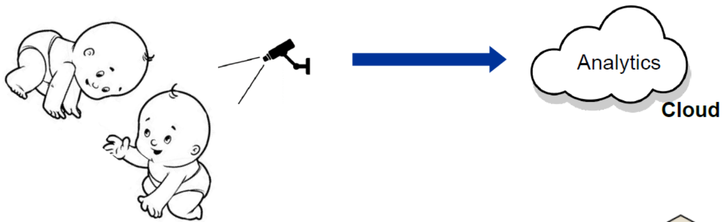


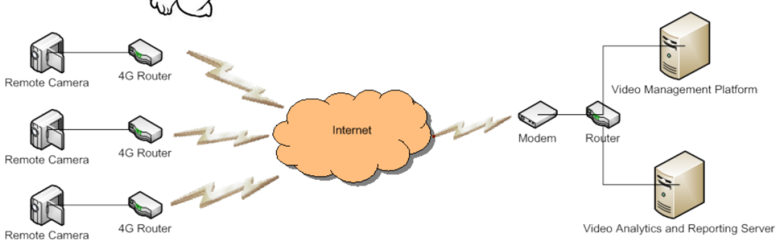
True positives (TP)	The data that is correctly classified by a model as positive instance of the concept being modelled.
False positives (FP)	The data that is classified as positive instance by the model, but in fact are known not to be
True negatives (TN)	The data correctly classified by the model as not being instances of the concept
False negatives (FN)	The data that is classified as not being instances, but are in fact know to be

- Deployment
  - How do you want to deploy? edge vs cloud? real-time vs. post-event?



  - Cloud
    - pros and cons

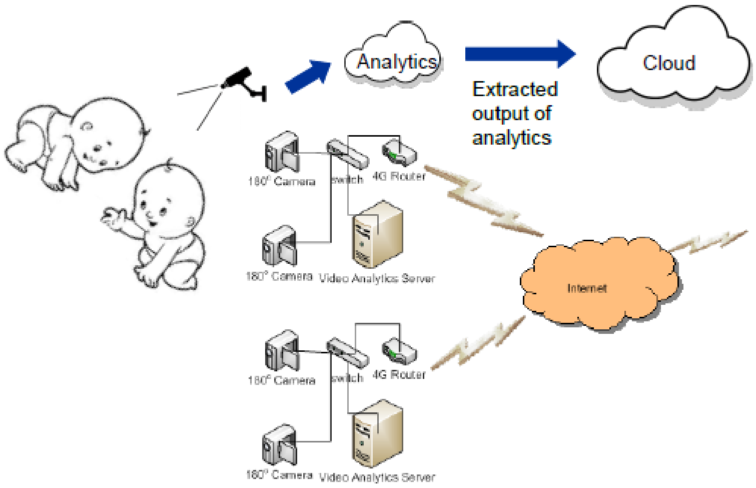




- Agility and Affordability: No capital investment of large-size infrastructures for analytical workloads.
- Data Analytics Platforms in Clouds: Leveraging cloud-enabled and ready platforms are fast and easy.
- Databases and Data Warehouses in Clouds: All kinds of database management systems and data warehouses in cloud speed up the process of data analytics.
- Enterprise-class Applications in Clouds: All kinds of customer-facing applications are cloud-enabled and deployed in highly optimized and organized cloud environments.
- Sensor/Device-to-Cloud Integration are available to transmit ground-level data to cloud storages and processing.

- Latency and Response time is often a critical part, especially when you deal with emergency procedure.
- Bandwidth Cost and Capacity: If you want to use a set of smart devices requiring each one to communicate certain bytes of data then you can quickly reach huge bandwidth requirements reaching Mbit/s or even Gbit/s at a gateway level.
- Security and Privacy: Transmitting device data over any open and public network is risky.
- Offline usages versus only-online usages: Pure cloud services do not allow offline usages. It is a major shortcoming since smart cities and industry 4.0 applications require a dual offline/online paradigm.

- Edge
    - pros and cons



Volume and Velocity: Processing and storing such huge amounts of data which is gathered in real-time.

Security: Devices can be located in sensitive environments, control vital systems or send private data.

Bandwidth: If devices constantly send the sensor and video data, it will hog the internet and cost a fortune.

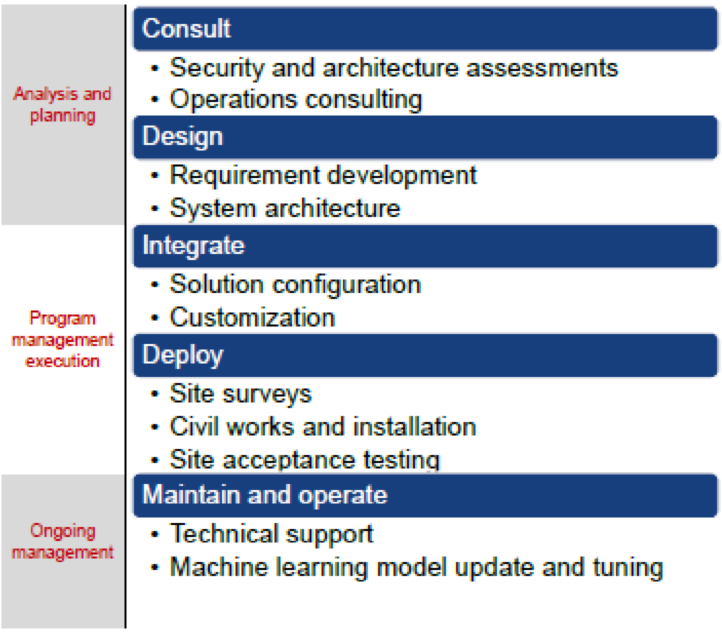
Real-time Data Capture, Storage, Processing, Analytics, Knowledge Discovery, Decision-making and Actuation.

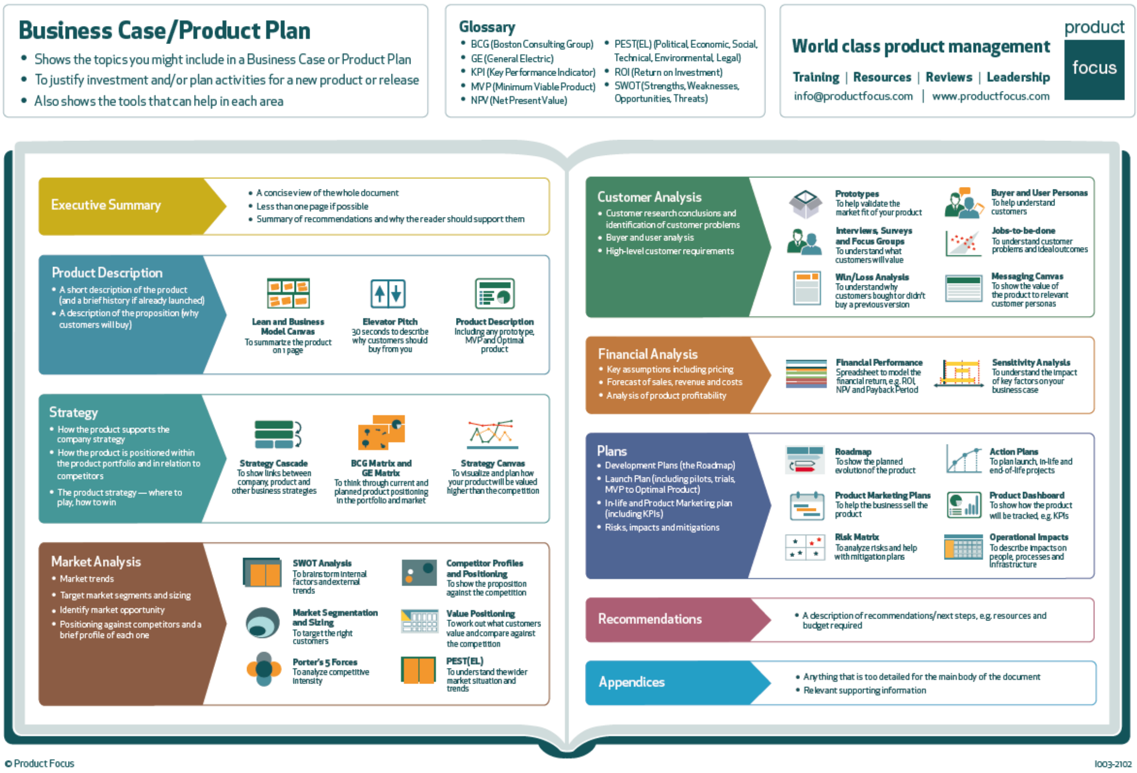
Less Latency and Faster Response.

- User concerns
    - What are the legal restrictions which may exist in your country?
    - Ethical considerations of data usage.
    - An example in CCTV security system

Stakeholders	People (being watched)	System owner	System operator
Acquisition	• Consent • Signage • Anonymity	• No data missing • Video properties	• None
Storage	• Secure storage	• Secure storage • Deletion after retention period	• None
Transmission	• Confidentiality • Integrity • Authenticity	• Confidentiality • Integrity • Authenticity	• None
Monitoring	• Privacy safeguards • Authorized access • Public access to their data	• Continuous monitoring • Authorized access	• Data freshness • Timestamping • Easy to search

- Business model






Vision
What is your motivation for creating the product?
Which positive change should it bring about?
Take the stress out of finding a meeting time for large groups
Target Group
Which market or market segment does the product address?
Who are the target customers and users?
Employees who plan meetings
Employees who attend meetings
Needs
Which problem does the product solve?
What benefit does it provide?
Easy to use
Simple
Work for a distributed team
Product
What product is it?
What makes it stand out?
Is it feasible to develop the product?
Syncs with calendar apps
Easy to use
Clean and simple design
Business Goals
How is the product going to benefit the company?
What are the business goals?
Decrease time spent planning meetings
Increase productivity
Competitors
Who are your main competitors?
What are their strengths and weaknesses?
Google Calendar
Robin
Email
Revenue Streams
How can you monetise your product and generate revenues?
Land and expand business model
Set up for SS
Cost Factors
What are the main cost factors to develop, market, sell, and service the product?
Tooling
Resourcing
Channels
How will you market and sell your product?
Do the channels exist today?
Website
Organic social
Email