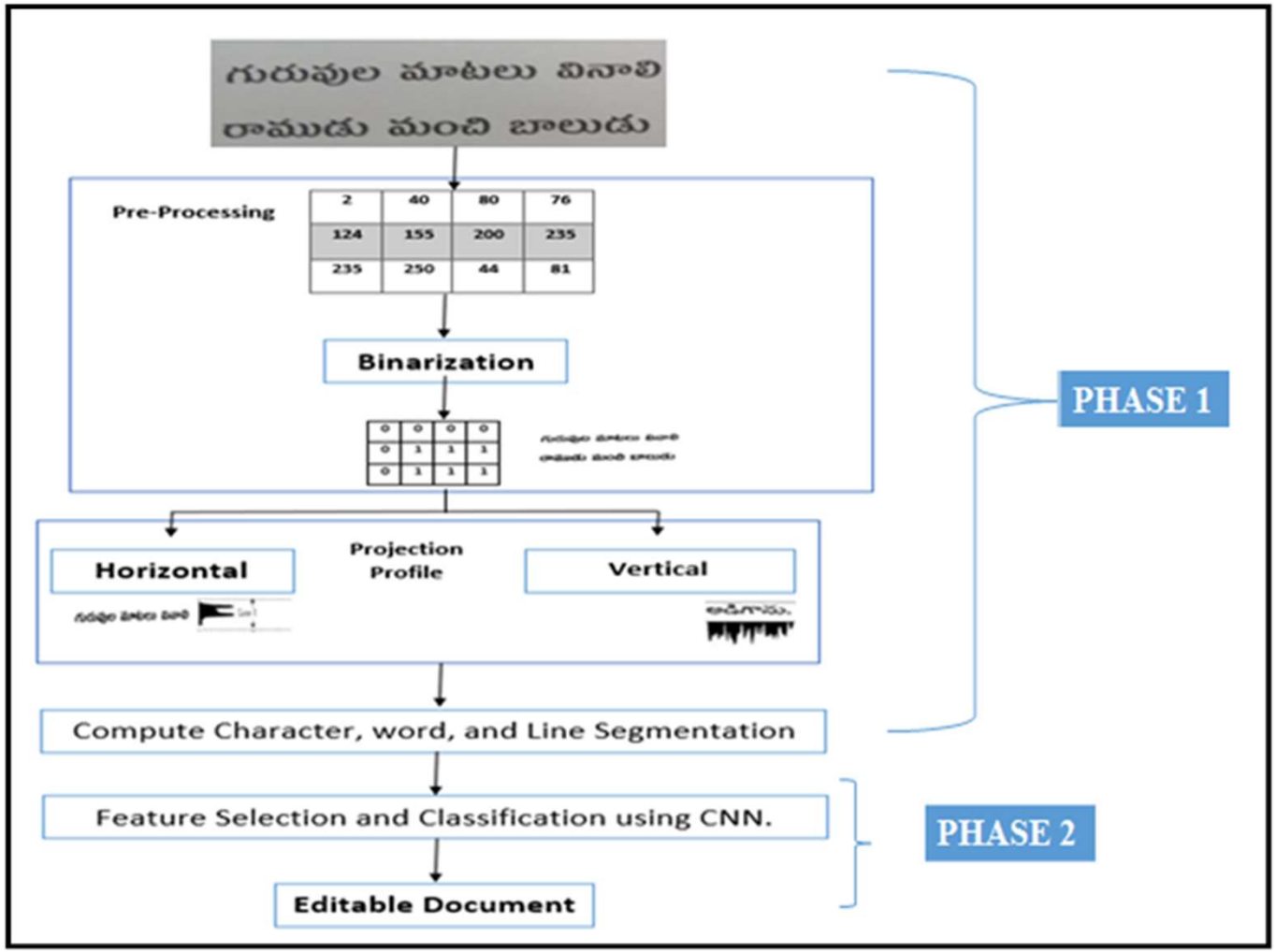
Day 2
Visual Computing FDP
Topic: Image Classification in CV

A picture is worth a thousand words.

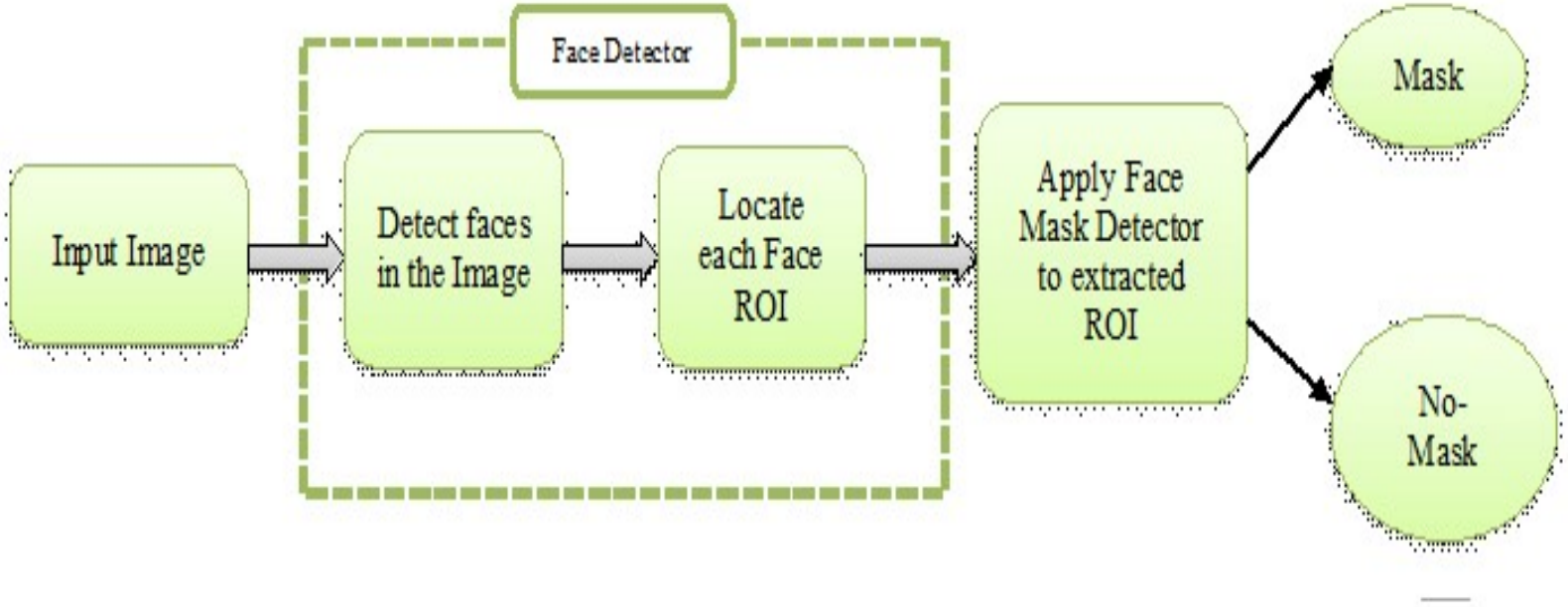
Research Problem Statement 1

Design a novel model to recommend
the text from scanned document



Research problem statement 2

Design a model to detect face mask



Research problem statement 3

Design an efficient model to generate a Human face using Diffusion models to identify Criminals

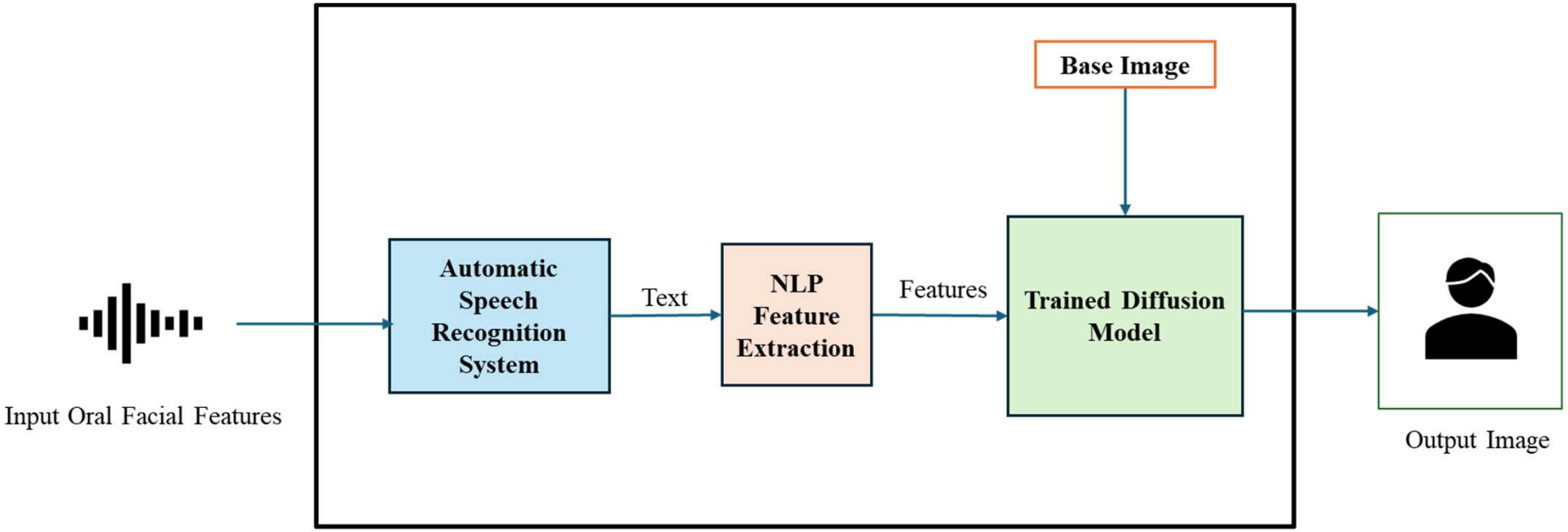
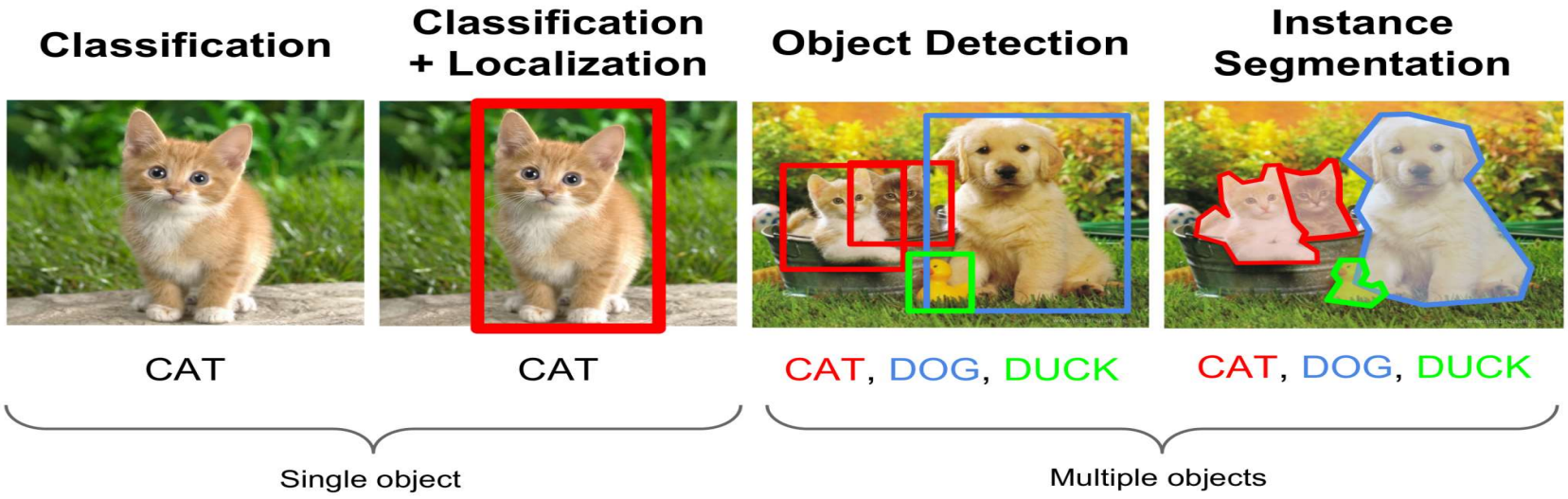


Figure 2: Human Face Generation Model Architecture

What is Image Classification?

- Image classification is the task of categorizing and assigning labels to groups of pixels or vectors within an image dependent on particular rules. The categorization law can be applied through one or multiple spectral or textural characterizations.



Supervised Classification

- Supervised classification is based on the idea that a user can select sample pixels in an image that are representative of specific classes and then direct the image processing software to use these training sites as references for the classification of all other pixels in the image.
- Supervised classification uses classification algorithms and regression techniques to develop predictive models.
- The algorithms include linear regression, logistic regression, neural networks, decision tree, support vector machine, random forest, naive Bayes, and k-nearest neighbor.

Unsupervised Classification

- Unsupervised classification is where the outcomes (groupings of pixels with common characteristics) are based on the software analysis of an image without the user providing sample classes.
- Unsupervised learning include cluster analysis, anomaly detection, n
- Two popular algorithms used for unsupervised image classification are 'K-mean' and 'ISODATA.'

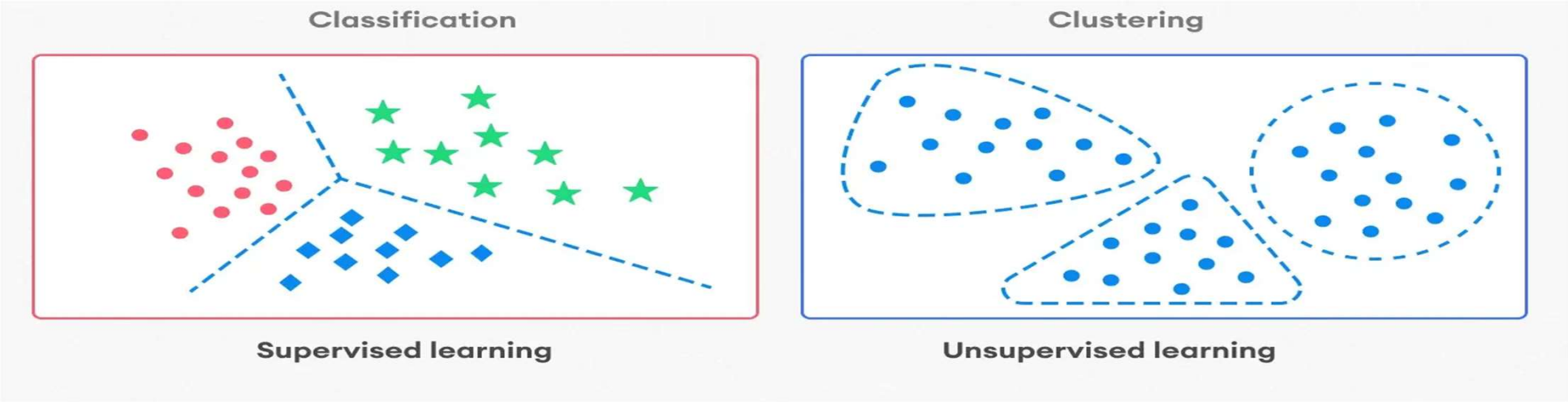


Image Classification in Machine Learning

Image Classification-Types

- Image classification is a supervised learning problem: define a set of target classes (objects to identify in images), and train a model to recognize them using labeled example photos.
- **Single –Label Classification :**
- Single-label classification is the most common classification task in supervised Image Classification.
- The output from the model is a vector with a length equal to the number of classes and value denoting the score that the image belongs to this class.
- A Softmax activation function is employed to make sure the score sums up to one and the maximum of the scores is taken to form the model's output.

While the Softmax initially seems not to provide any value to the prediction as the maximum probable class does not change after applying it, it helps to bound the output between one and zero, helping gauge the model confidence from the Softmax score.

Softmax

$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

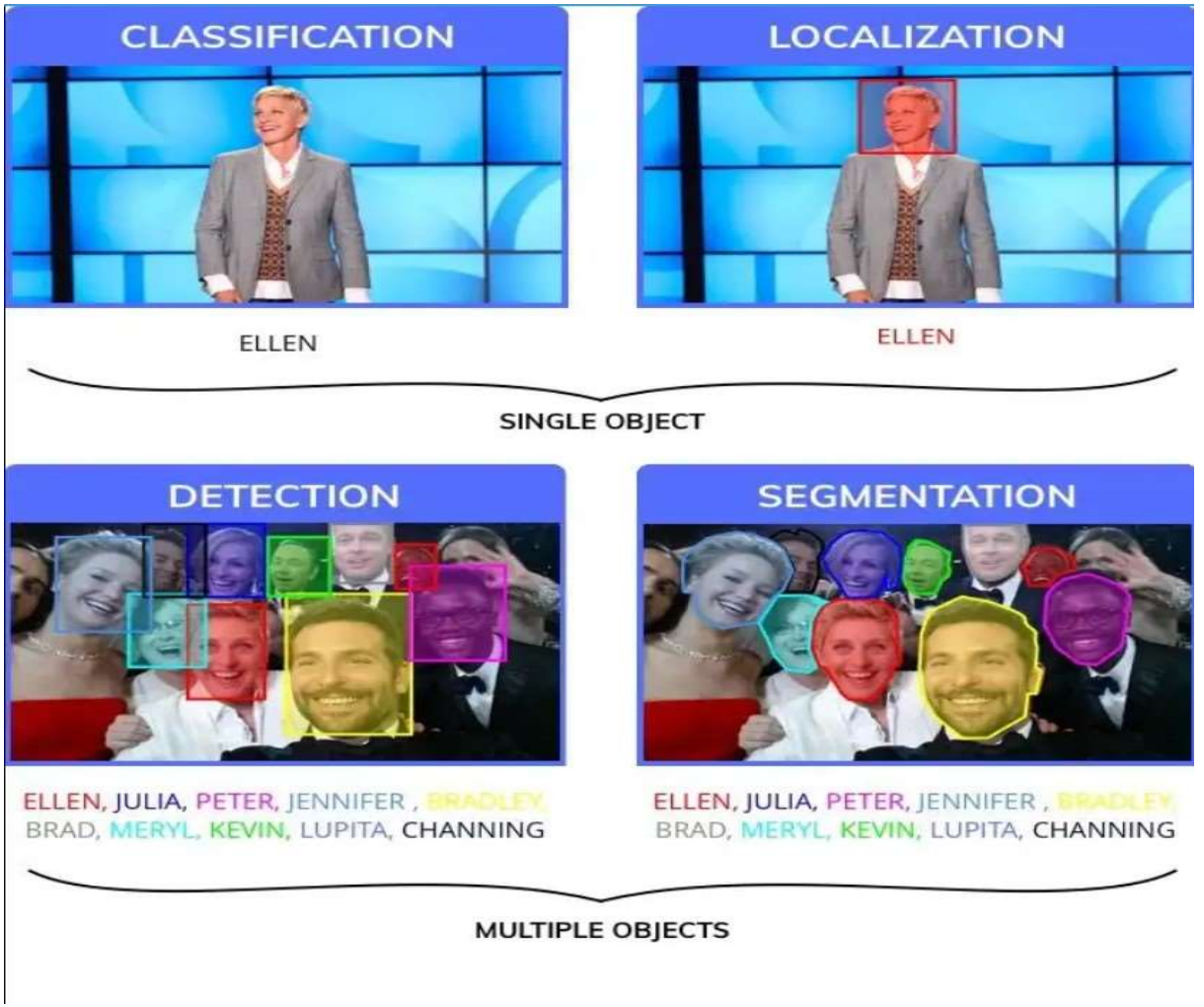
Multi-label Classification

- Multi-label classification is a classification task where each image can contain more than one label, and some images can contain all the labels simultaneously.

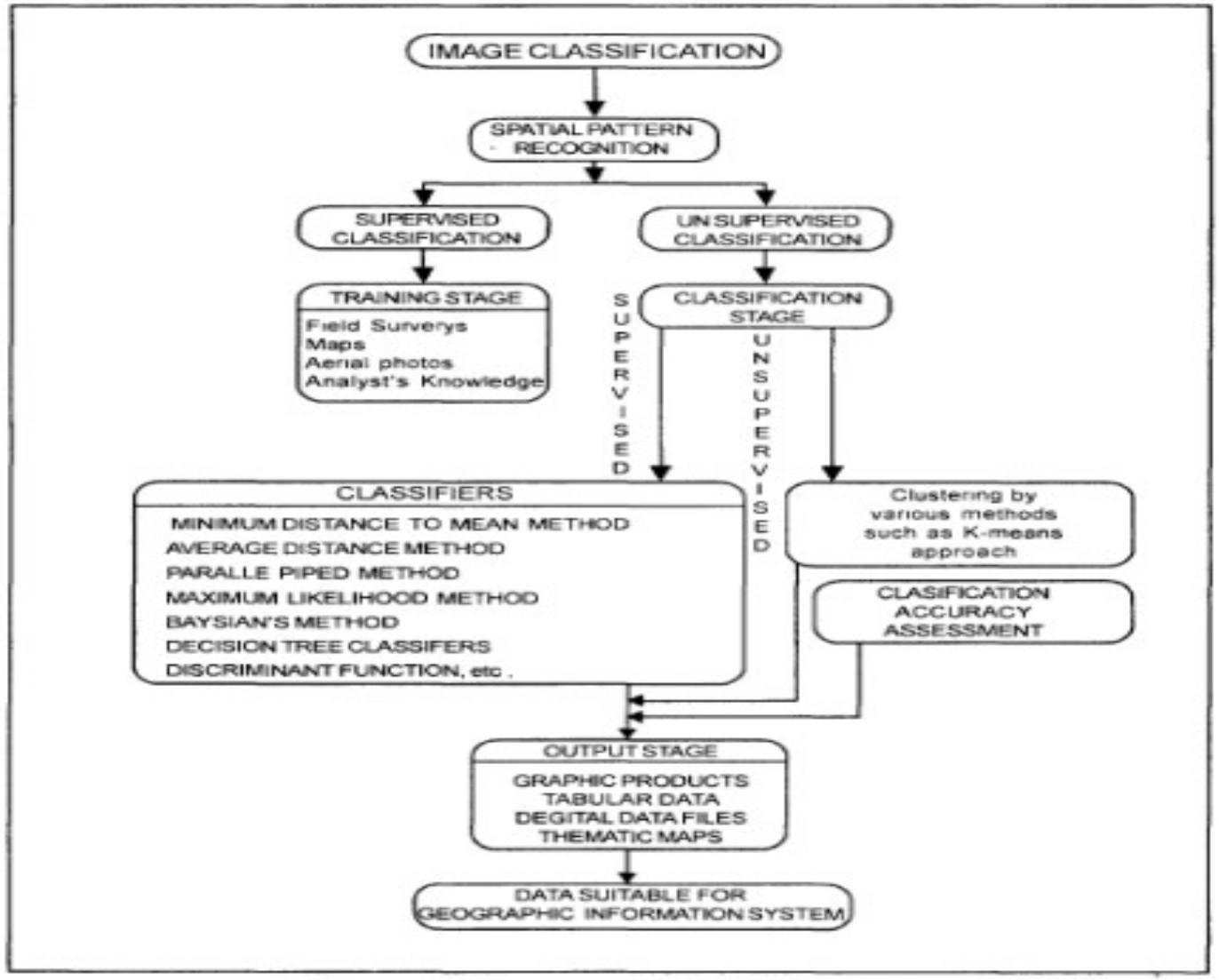
- While this seems similar to single-label classification in some respect, the problem statement is more complex compared to single-label classification.

- Multi-label classification tasks popularly exist in the medical imaging domain where a patient can have more than one disease to be diagnosed from visual data in the form of X-rays.

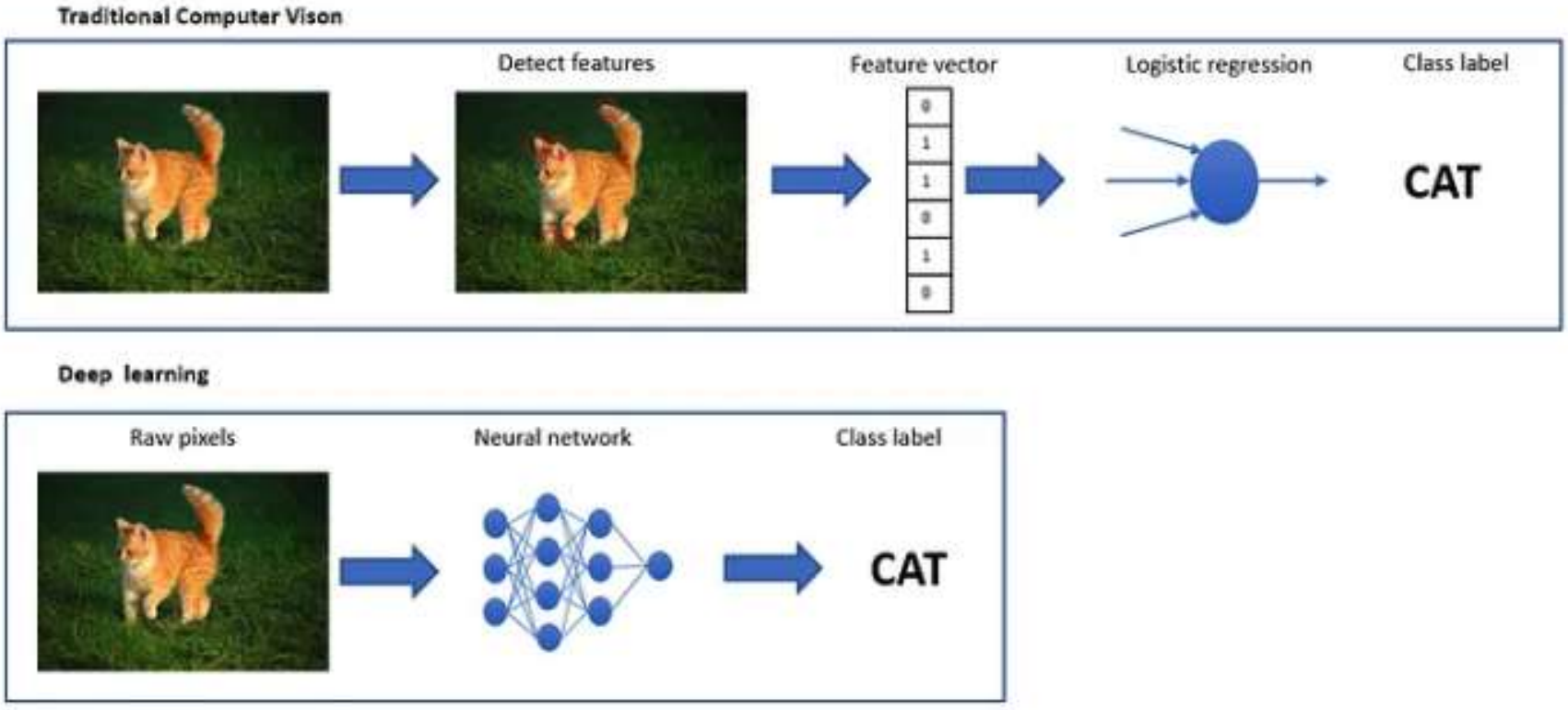
Classification, Localization, Detection and Segmentation



Flow chart of Image Classification

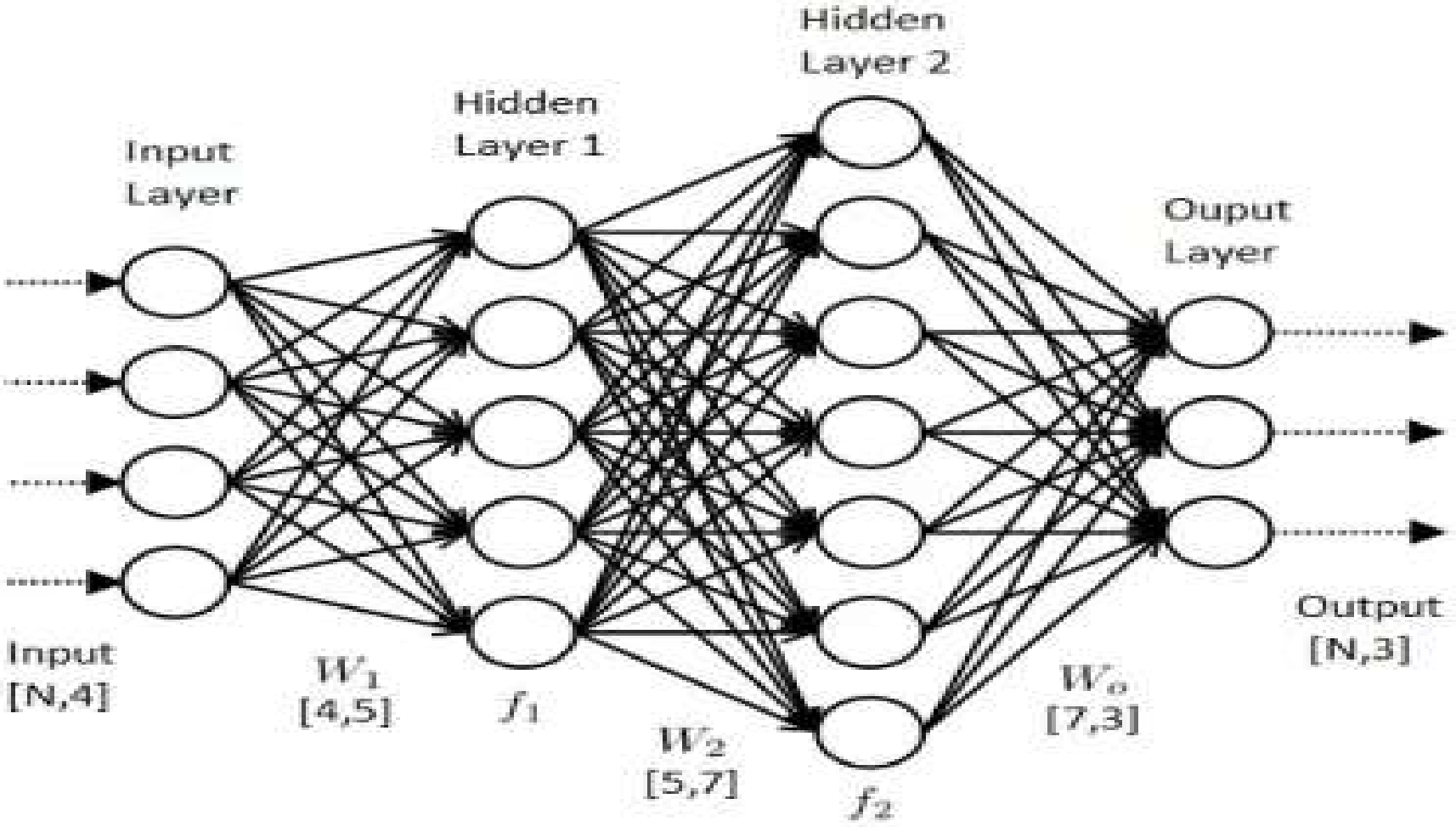


Traditional method for Classification

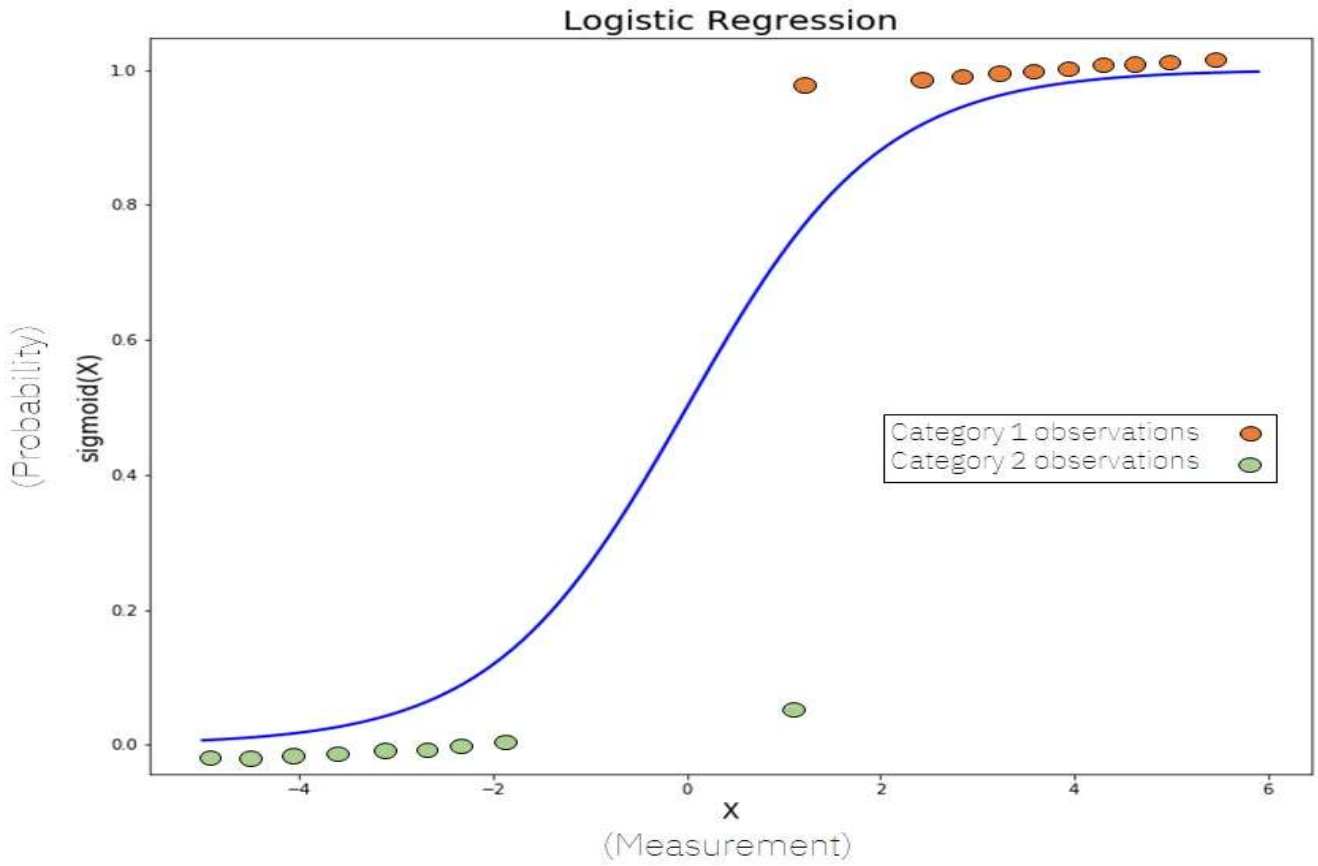


Source

ANN

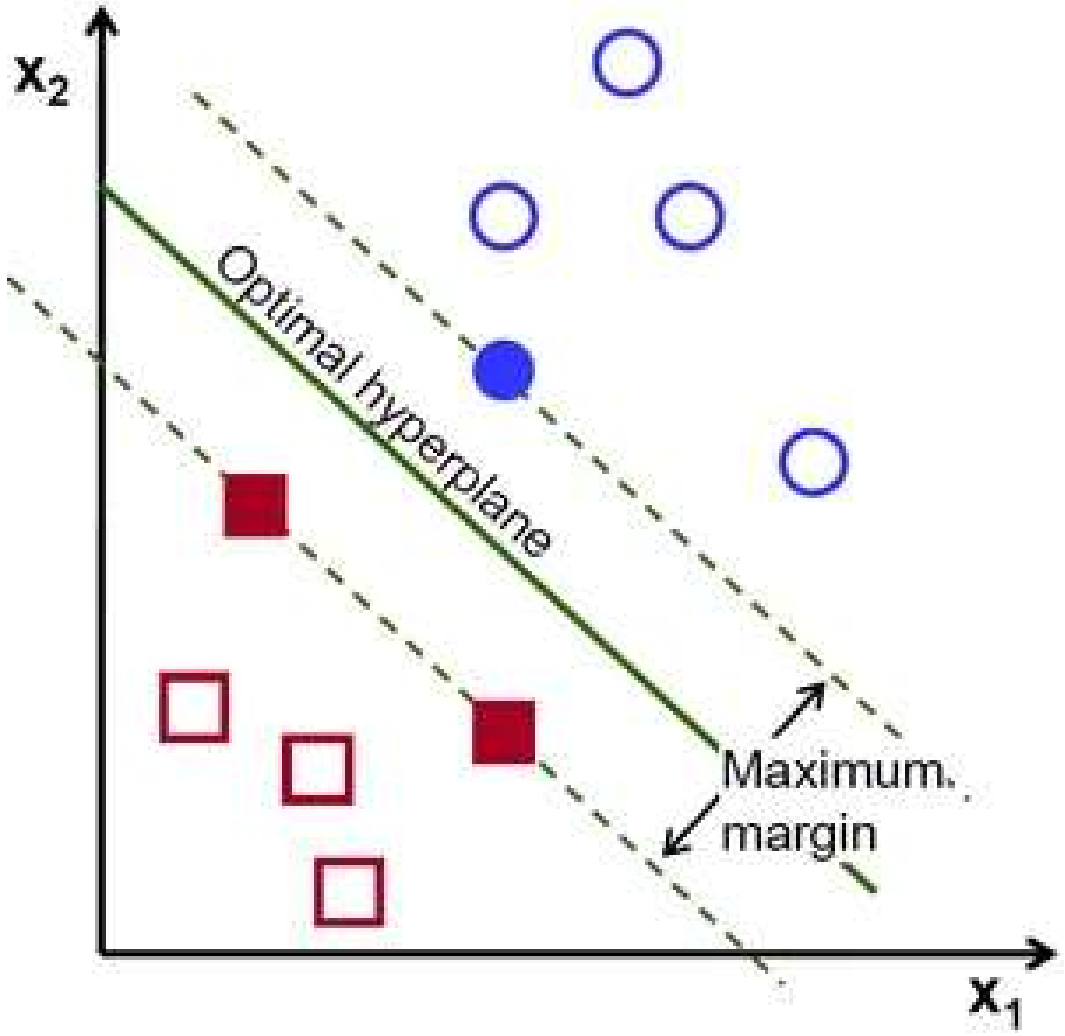

Input Layer
Hidden Layer 1
Hidden Layer 2
Ouput Layer
Input [N,4]
W_1 [4,5]
f_1
W_2 [5,7]
f_2
W_o [7,3]
Output [N,3]

Logistic Regression



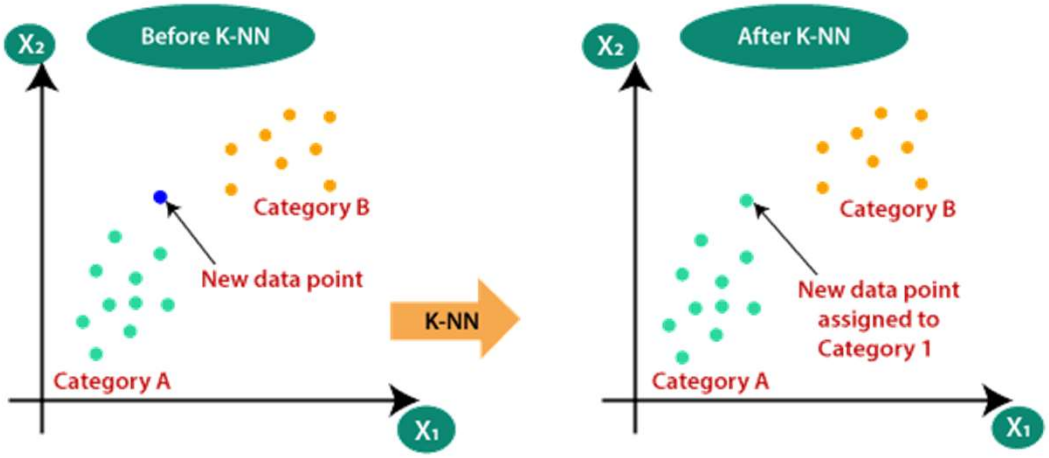
SVM

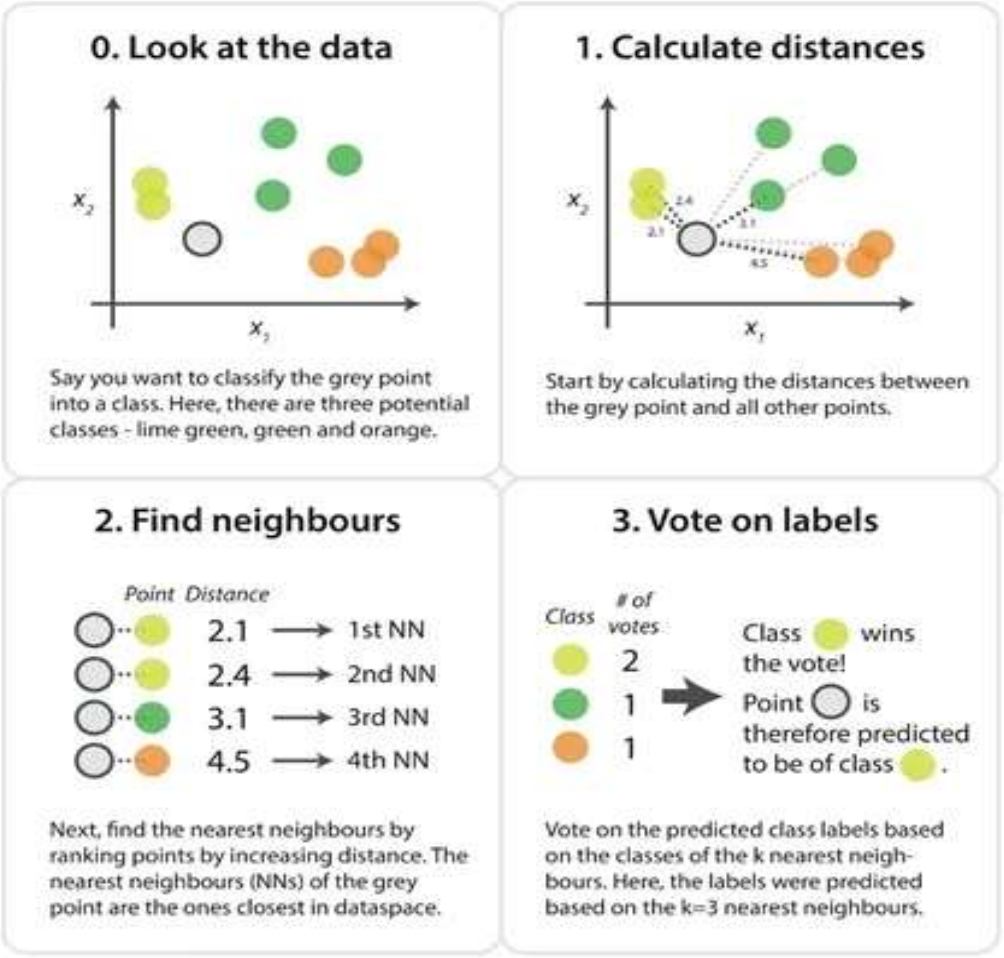
Support vector machines (SVM) are powerful yet flexible supervised machine learning algorithms which are used both for classification and regression.



K-NN

K-Nearest Neighbor is a non-parametric method used for classification and regression.





Random forest

Random forest is a supervised learning algorithm which is used for both classification as well as regression.

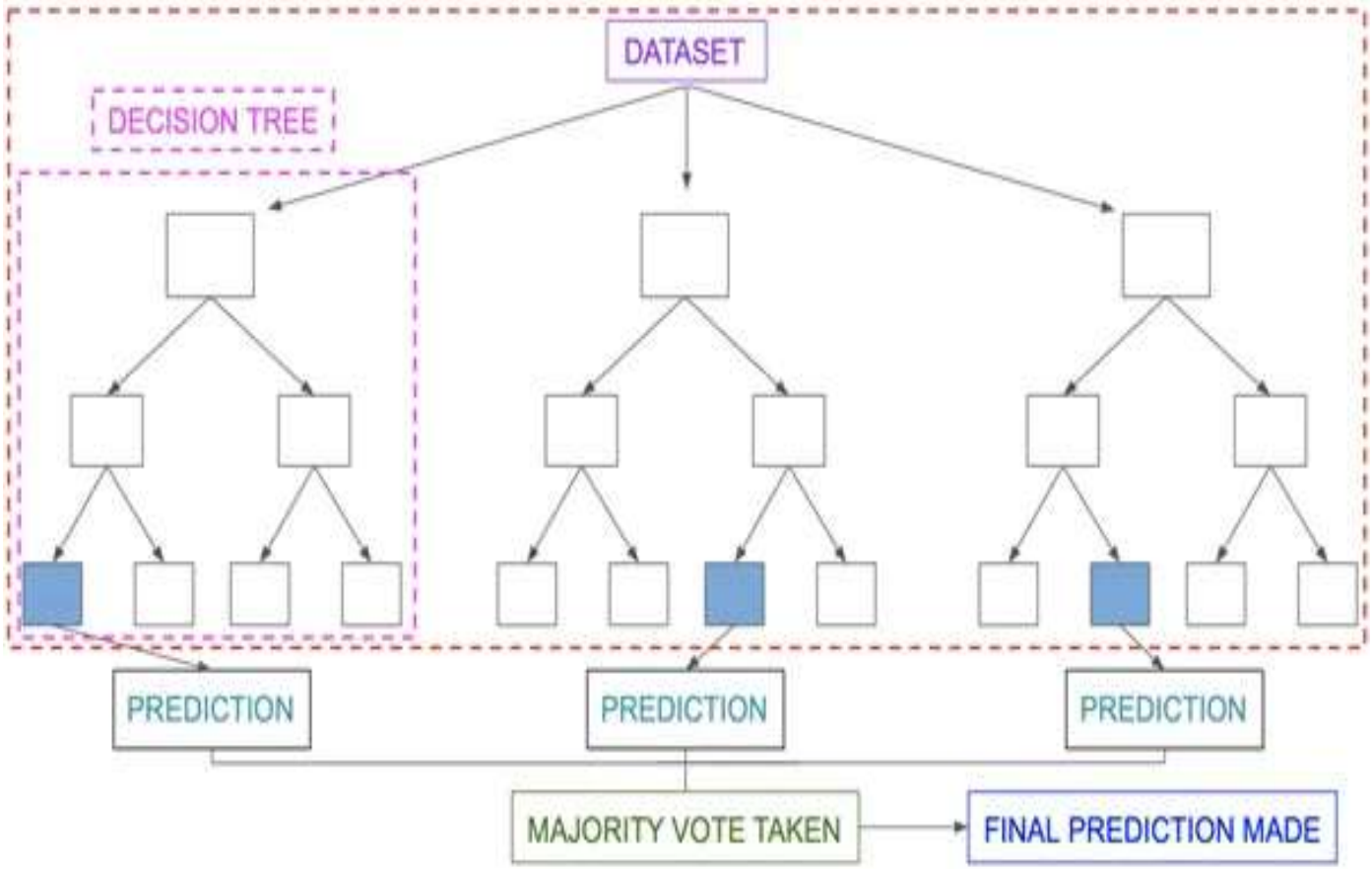
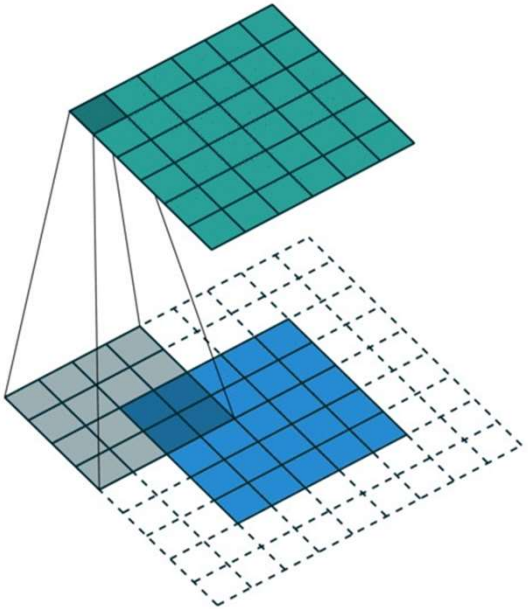


Image Classification using DL

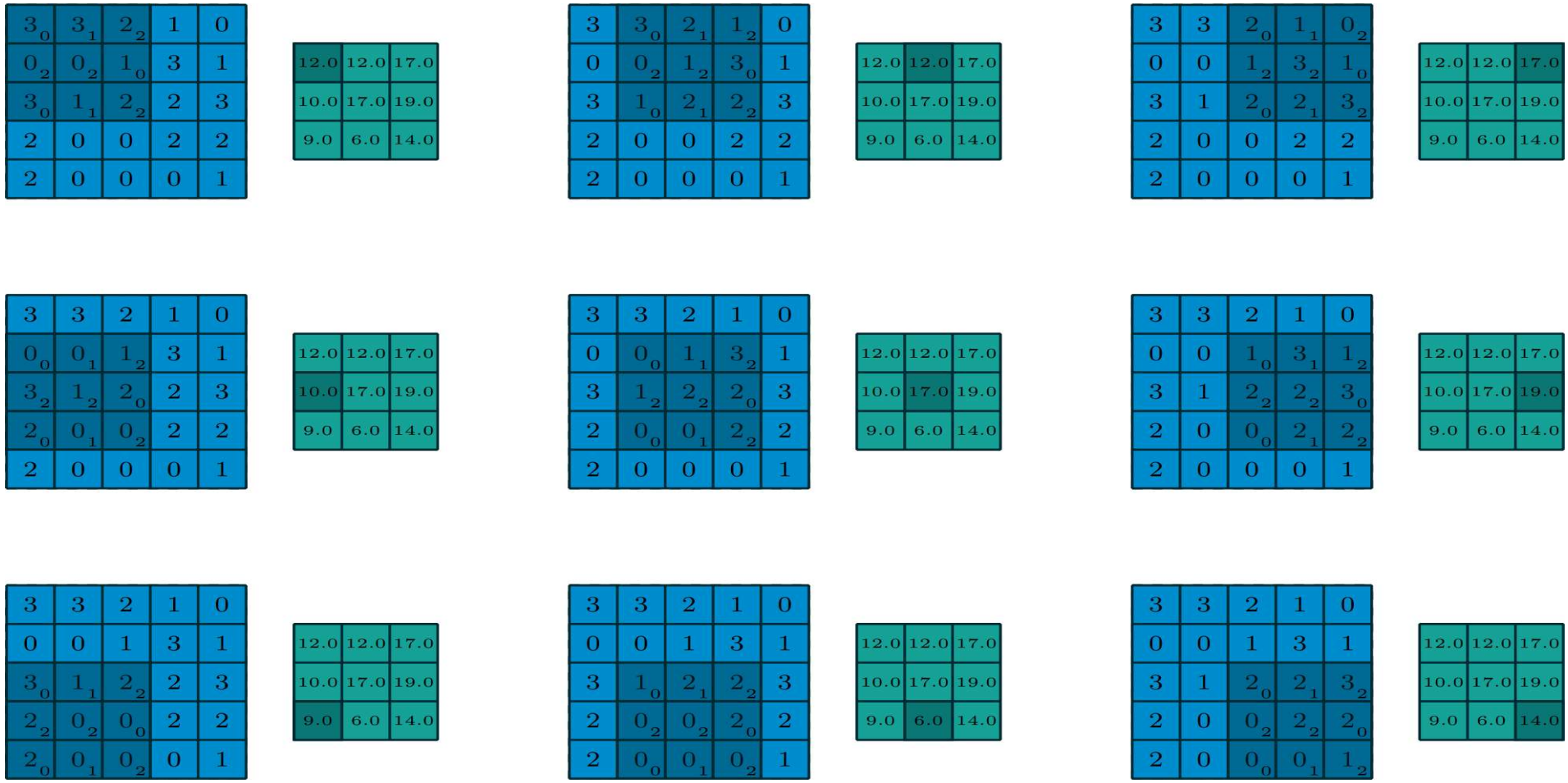
Classification models: CNN

- In modern computer vision, the *gold-standard* model architecture to use is a convolutional neural network.
- The convolution operation slides a matrix, called kernel, on input while performing pair-wise matrix multiplication and summing up the result. The following gif shows this process.



0	1	2
2	2	0
0	1	2

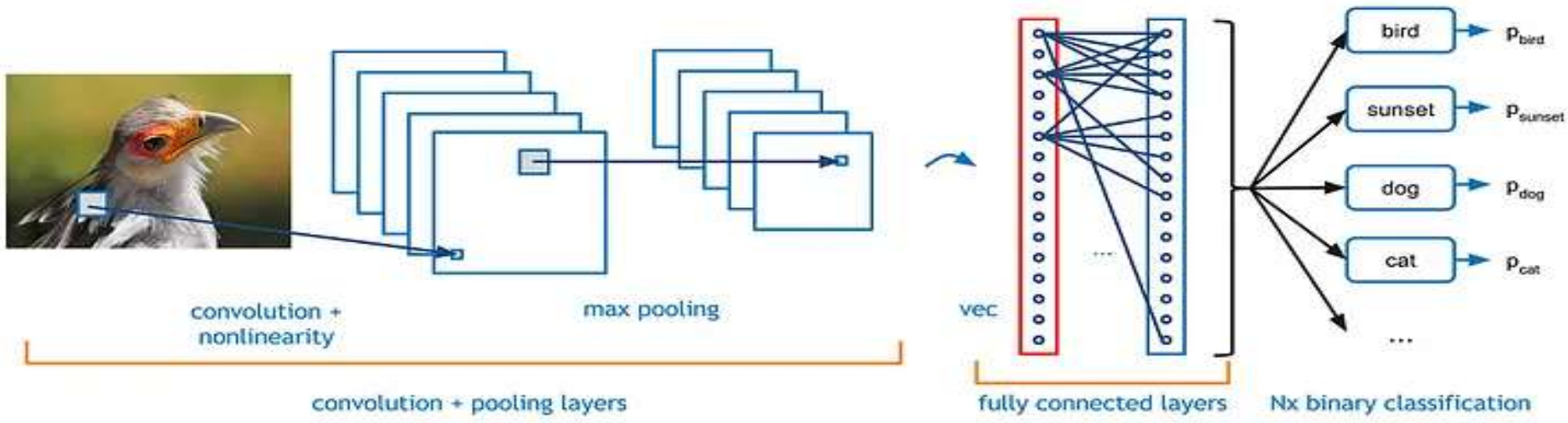
Convolution process



CNN Architecture for Image Classification

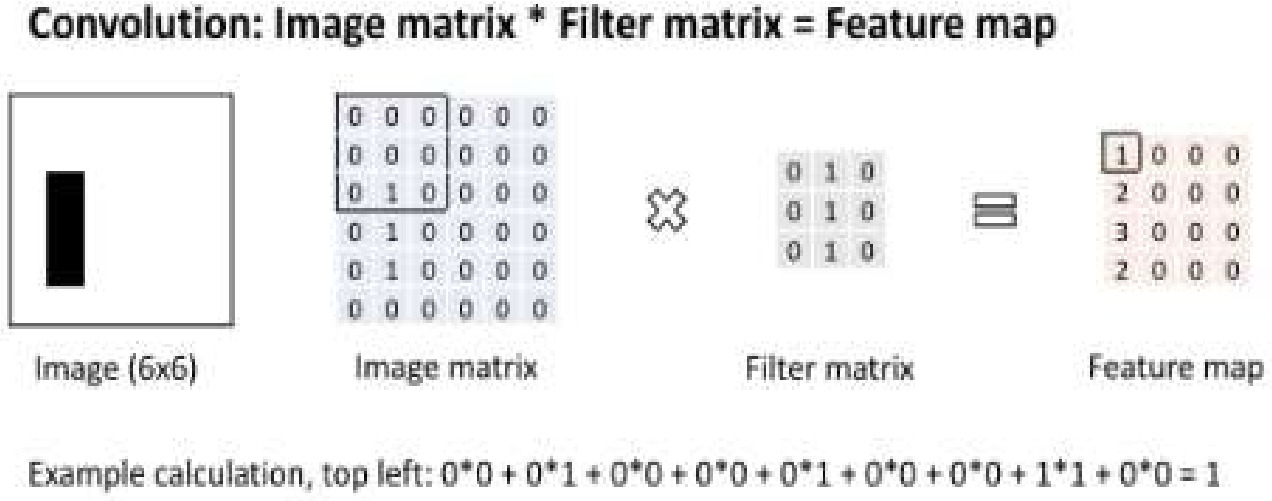
Basic CNN architecture for Classification:

- The CNN architecture for image classification includes
- Convolutional layers,
- Max-pooling layers, and
- Fully connected layers.



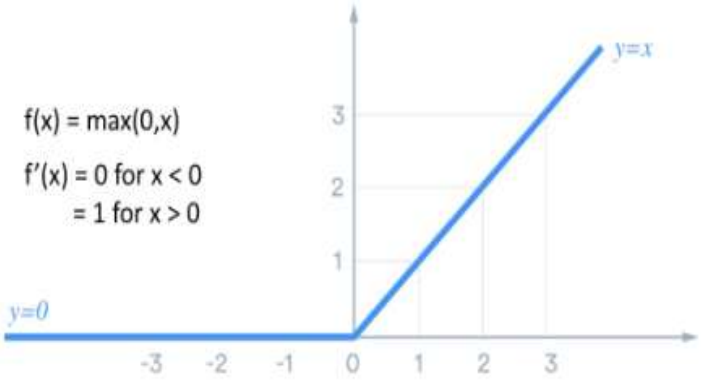
Convolution layer

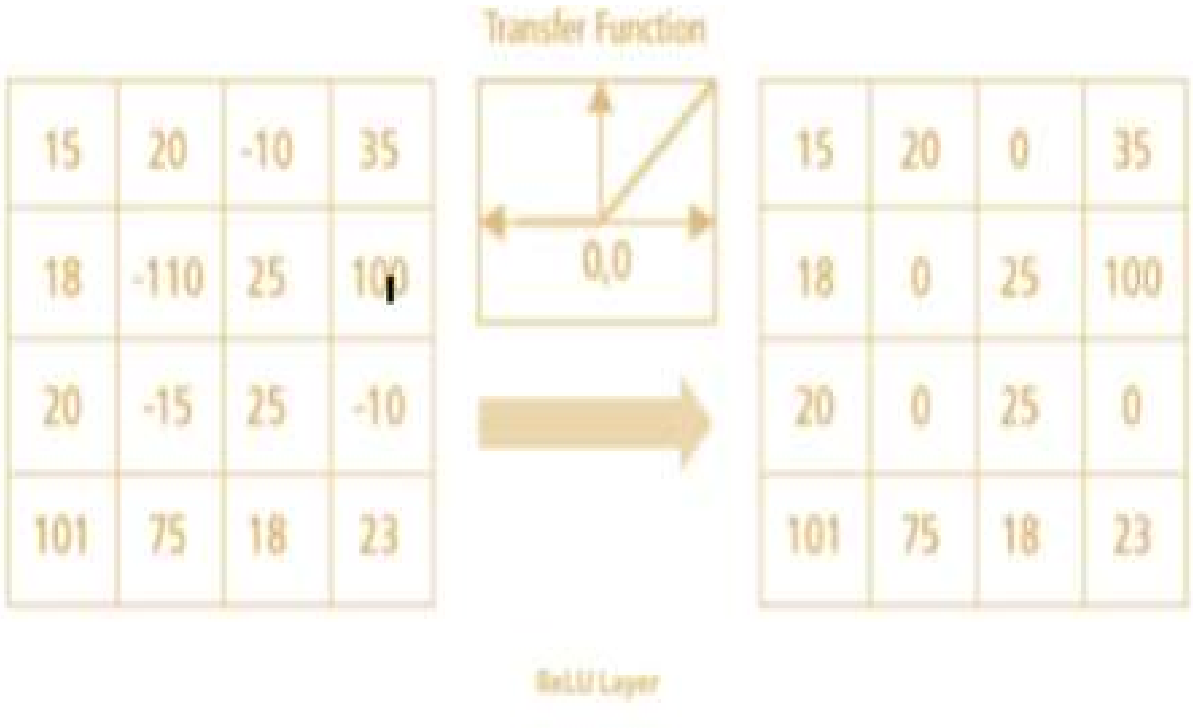
- Convolutional layers apply a set of filters to the input image to extract important features, such as edges, lines, textures, and shapes. Each filter is a small matrix of weights that is convolved with the input image to produce a feature map. Each feature map represents a particular feature or pattern that the CNN has learned to detect in the input image.



ReLU layer

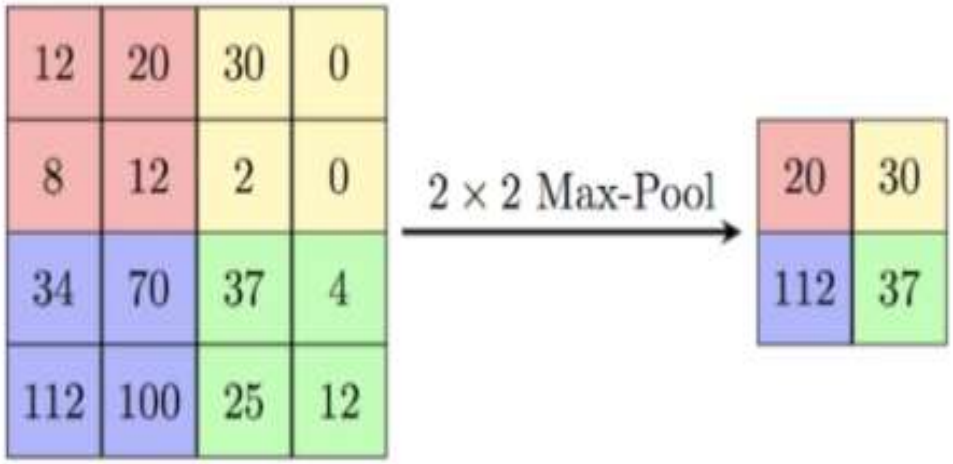
- The output of convolutional layer is passed to, let's say, rectified linear unit (ReLU) activation function. You can also use other activation functions as well. This introduces non-linearity into the model.



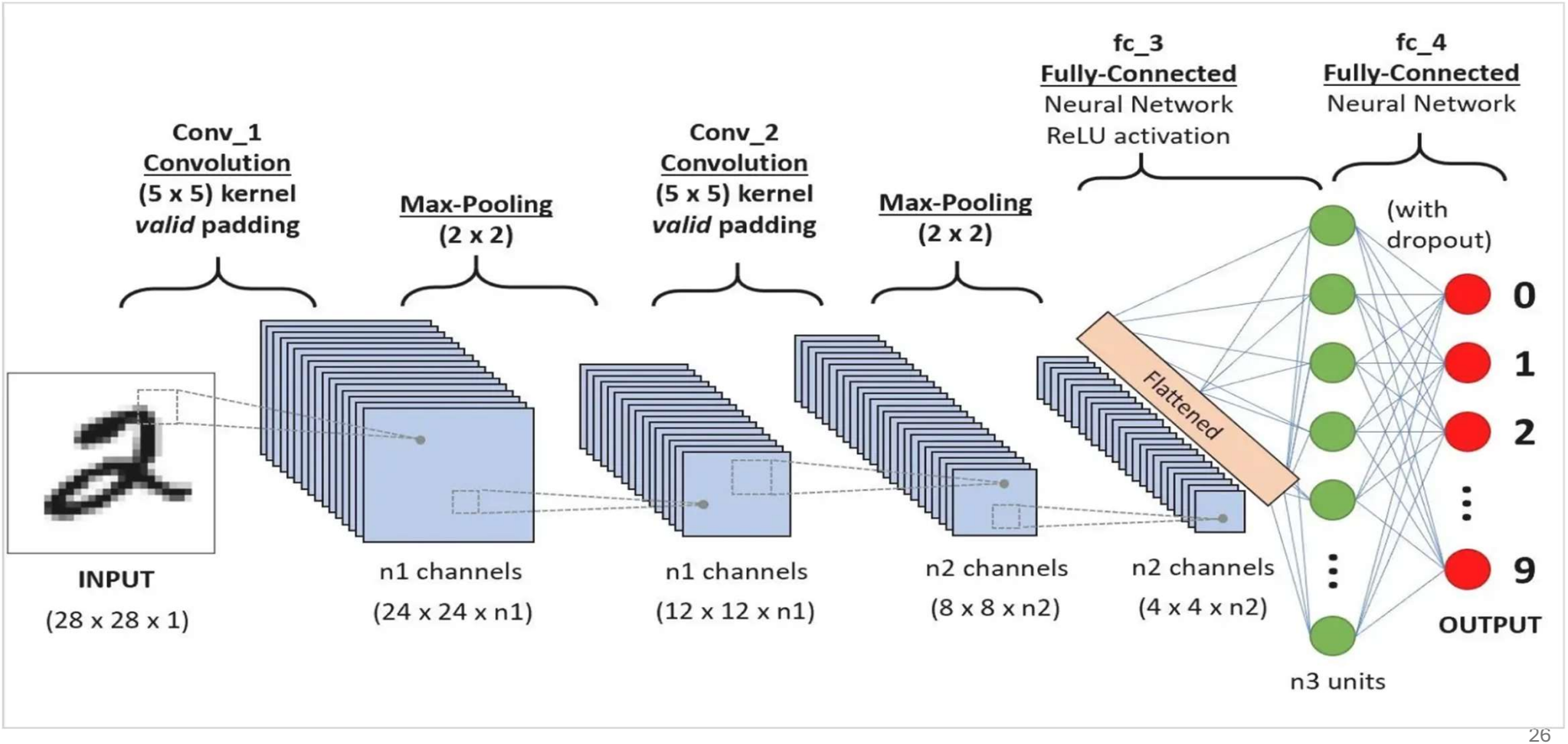


Max Pooling layer

- This layer down-samples the feature maps produced by the second convolutional layer, further reducing their dimensionality



Basic CNN architecture for Classification



Applications

- Medical and Healthcare
- Manufacturing
- Security
- Agriculture
- Autonomous Vehicles

Image Representation

Representation and Description

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data that constitutes either the boundary of a region or all the points in the region itself.
Description deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

- Representation
  - Boundary Based (external characteristics)
    - Used when interested in the shape of the object
  - Region Based (internal characteristics)
    - Used when interested in surface reflectance property such as color, texture
- Description
  - Boundary Based
  - Region Based

Representation and Description

- A segmented region can be represented by { boundary pixels / internal pixels }
- When shape is important, a boundary (external) representation is used
- When colour or texture is important, an internal representation is used
- The description of a region is based on its representation, for example a boundary can be described by its length
- The features selected as descriptors are usually required to be as insensitive as possible to variations in (1) scale, (2) translation and (3) rotation, that is the features should be scale, translation and rotation invariant

Methods for Representation and Description

1. Representation

1. Boundary (Border) Following
2. Chain Codes
3. Polygonal Approximations Using Minimum-Perimeter Polygons
4. Other Polygonal Approximation Approaches
5. Signatures

11.1.7 Skeletons

2. Boundary Descriptors

1. Some Simple Descriptors
2. Shape Numbers

3. Regional Descriptors

1. Some Simple Descriptors
2. Topological Descriptors
3. Texture

Representation:

Boundary Following:

Several of the algorithms discussed in this chapter require that the points in the boundary of a region be ordered in a clockwise (or counterclockwise) direction. Consequently, we begin our discussion by introducing a boundary-following algorithm whose output is an *ordered* sequence of points. We assume

- We assume
  1. That we are working with binary images in which object and background points are labeled 1 and 0, respectively
  2. That images are padded with a border of 0s to eliminate the possibility of an object merging with the image border
- For convenience, we limit to single regions

- **Representation: Boundary Flowing Algorithm**

Moore boundary tracking algorithm (output is ordered sequence of points)

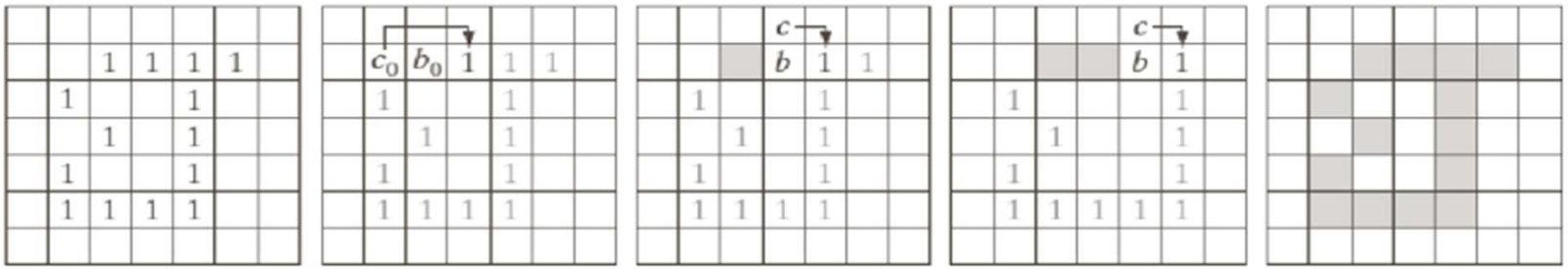
(1) Let the starting point b_0 be the uppermost, leftmost point in the image with label 1, and c_0 the west neighbour of b_0. Note that c_0 is always a background point. Examine the 8-neighbours of b_0, starting at c_0 and proceeding in a clockwise direction. Let b_1 denote the first neighbour encountered whose value is 1 and let c_1 be the (background) point immediately preceeding b_1 in the sequence. Store the locations of b_0 and b_1 for use in Step 5.

(2) Let $b = b_1$ and $c = c_1$.

(3) Let the 8-neighbours of b, starting at c and proceeding in a clockwise direction, be denoted by $n_1, n_2, \ldots, n_8$. Then find the first n_k labelled 1.

(4) Let $b = n_k$ and $c = n_{k-1}$.

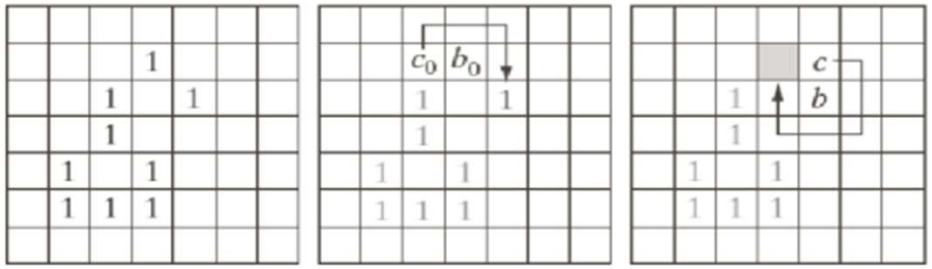
(5) Repeat Steps 3 and 4 until $b = b_0$ and the next boundary point found is b_1. The sequence of b points found when the algorithm stops constitutes the set of ordered boundary points.



a b c d e

FIGURE 11.1 Illustration of the first few steps in the boundary-following algorithm. The point to be processed next is labeled in black, the points yet to be processed are gray, and the points found by the algorithm are labeled as gray squares.

The algorithm should NOT stop the first time that b_0 is encountered again, as illustrated by the following example:

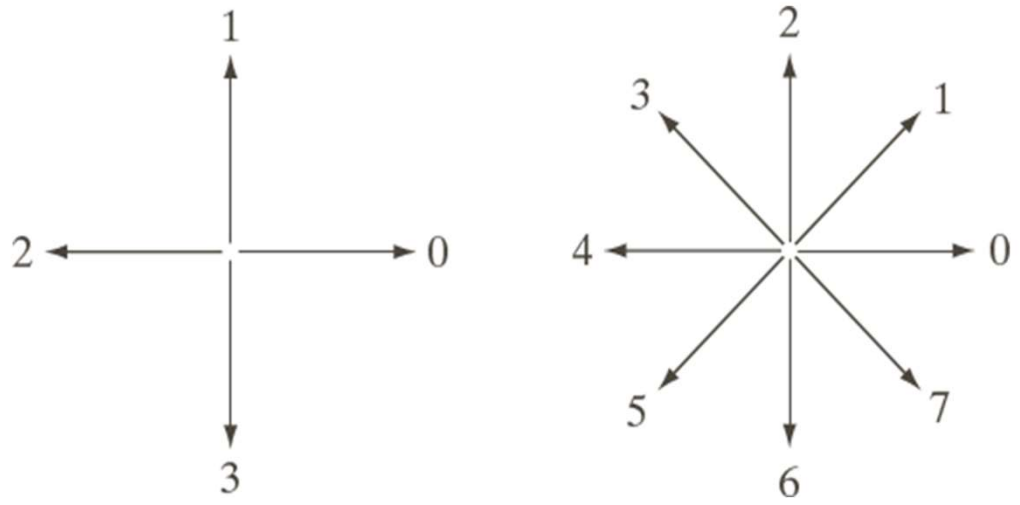


a b c

FIGURE 11.2 Illustration of an erroneous result when the stopping rule is such that boundary-following stops when the starting point, b_0, is encountered again.

Representation: Chain codes

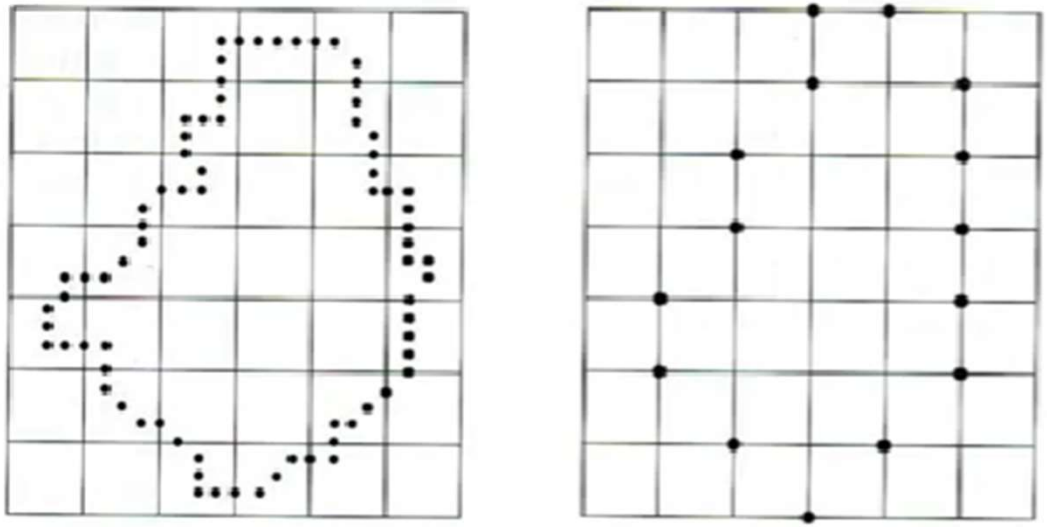
Chain codes are used to represent a boundary by a connected sequence of straight-line segments of specified length and direction.



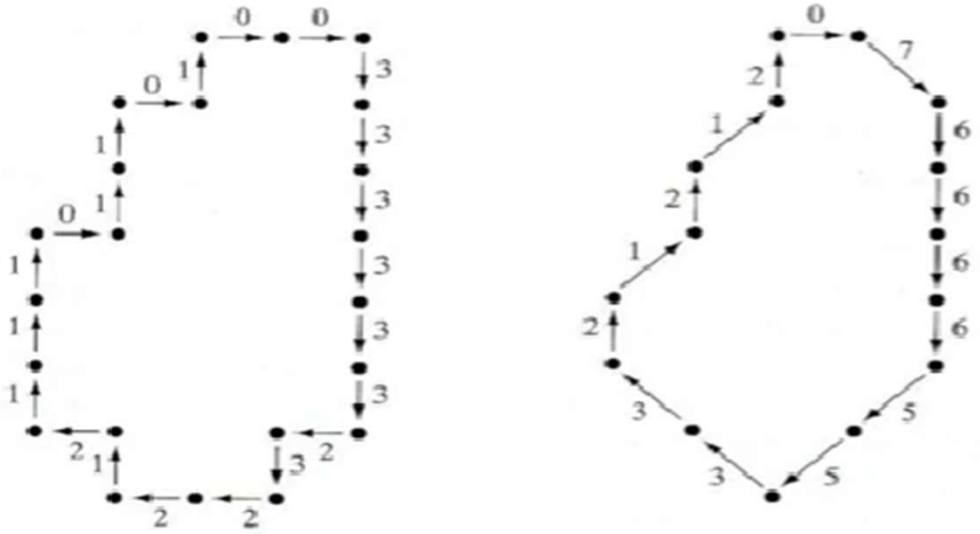
a b

FIGURE 11.3 Direction numbers for (a) 4-directional chain code, and (b) 8-directional chain code.

Representation: Chain codes example



a. Digital boundary with resampling grid superimposed
b. Result of resampling



(c) 4-directional chain code
(d) 8-directional chain code

The boundary representation in Fig.(c) is the chain code 0033 . . . 01, and in Fig. (d) it is the code 0766…12

The accuracy of the resulting code representation depends on the spacing of the sampling grid

The chain code of a boundary depends on the starting point

Example

Chain code is 10103322

Normalized with first difference 3133030

If we treat the code as a circular sequence to normalize with respec to the starting point,the result is 33133030.

Chain code depends on starting point

Normalization: consider the code to be circular and choose the starting point in such a way that the sequence represents the smallest integer

- This difference is obtained by counting the number of direction changes (in a counterclockwise direction) that separate two adjacent elements of the code
- For instance, the first difference of the 4-direction chain code 10103322 is 3133030
- If we elect to treat the code as a circular sequence, then the first element of the difference is computed by using the transition between the last and first components of the chain

Chain code disadvantages

- This method generally is unacceptable for two principal reasons
- (1) The resulting chain of codes tends to be quite long

and

- (2) any small disturbances along the boundary due to noise or imperfect segmentation cause changes in the code that may not be related to the shape of the boundary

Representation: Polygon Approximation using Minimum perimeter polygon(MPP)

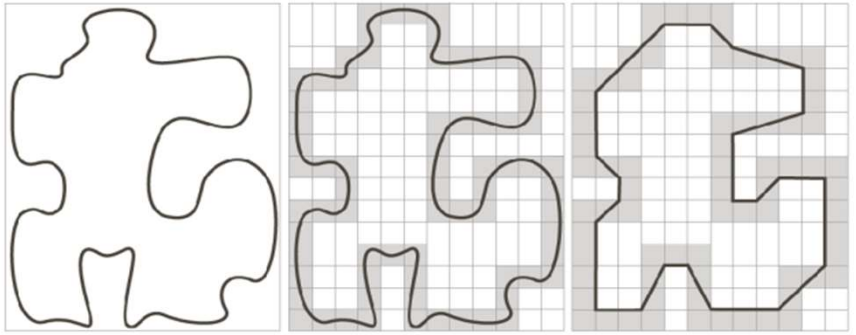
A digital boundary can be approximated with arbitrary accuracy by a polygon

The approximation is exact when the number of segments in the polygon is equal to the number of points in the boundary

So that each pair of adjacent points defines a segment in the polygon

Cellular complex $\equiv$ set of cells enclosing digital boundary

Imagine the boundary as a "rubber band" and allow it to shrink...



a b c

FIGURE 11.6 (a) An object boundary (black curve). (b) Boundary enclosed by cells (in gray). (c) Minimum-perimeter polygon obtained by allowing the boundary to shrink. The vertices of the polygon are created by the corners of the inner and outer walls of the gray region.

The maximum error per grid cell is $\sqrt{2}d$, where d is the dimension of a grid cell

MPP algorithm (READ)

Representation: Other polygonal Approximation methods-Merging Technique

Merging techniques

(1) Consider an arbitrary point on the boundary

(2) Consider the next point and fit a line through these two points: $E = 0$ (least squares error is zero)

(3) Now consider the next point as well, and fit a line through all three these points using a least squares approximation. Calculate E

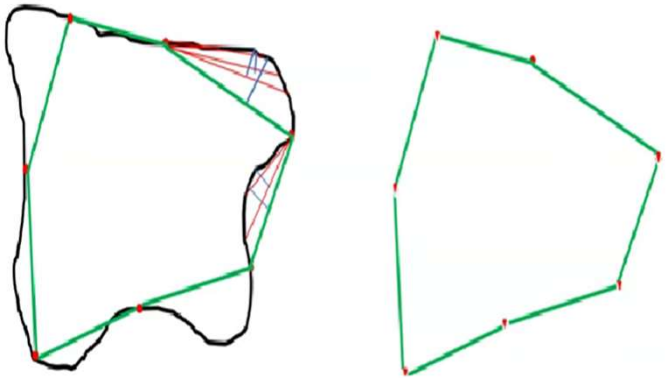
(4) Repeat until $E > T$

(5) Store a and b of $y = ax + b$, and set $E = 0$

(6) Find the following line and repeat until all the edge pixels were considered

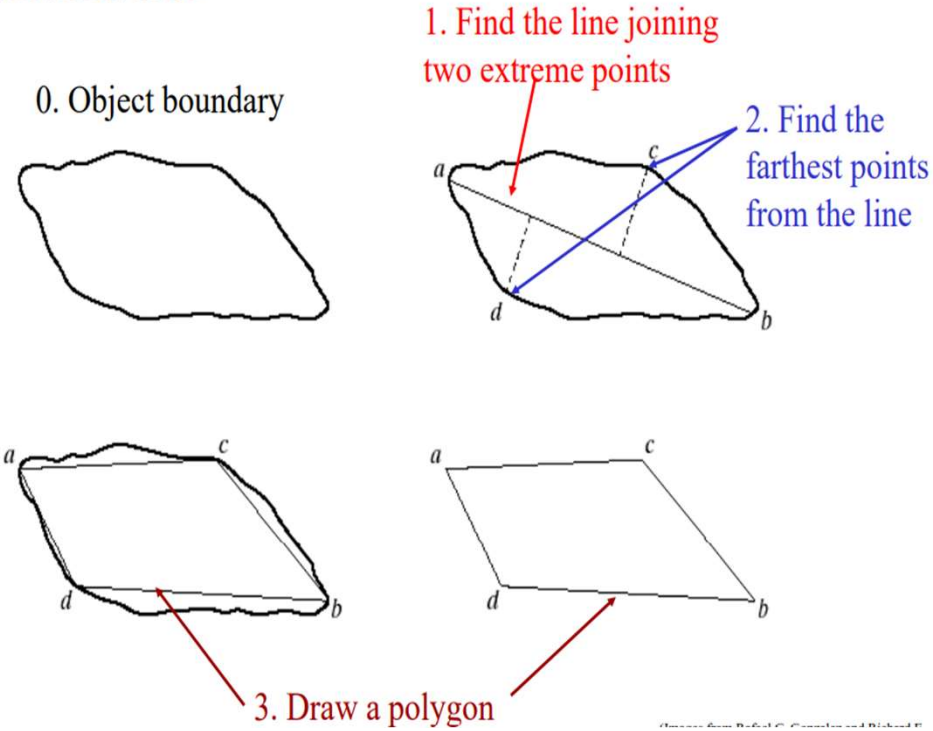
(7) Calculate the vertices of the polygon, that is where the lines intersect

Problems: Vertices do not always correspond to actual corners in the boundary: a new line is started after we have already turned, that is T is exceeded too late



Splitting techniques

- **Joint the two furthest points on the boundary $\longrightarrow$ line ab**
- **Obtain a point on the upper segment, that is c and a point on the lower segment, that is d, such that the perpendicular distance from these points to ab is as large as possible**
- **Now obtain a polygon by joining c and d with a and b**
- **Repeat until the perpendicular distance is less than some predefined fraction of ab**

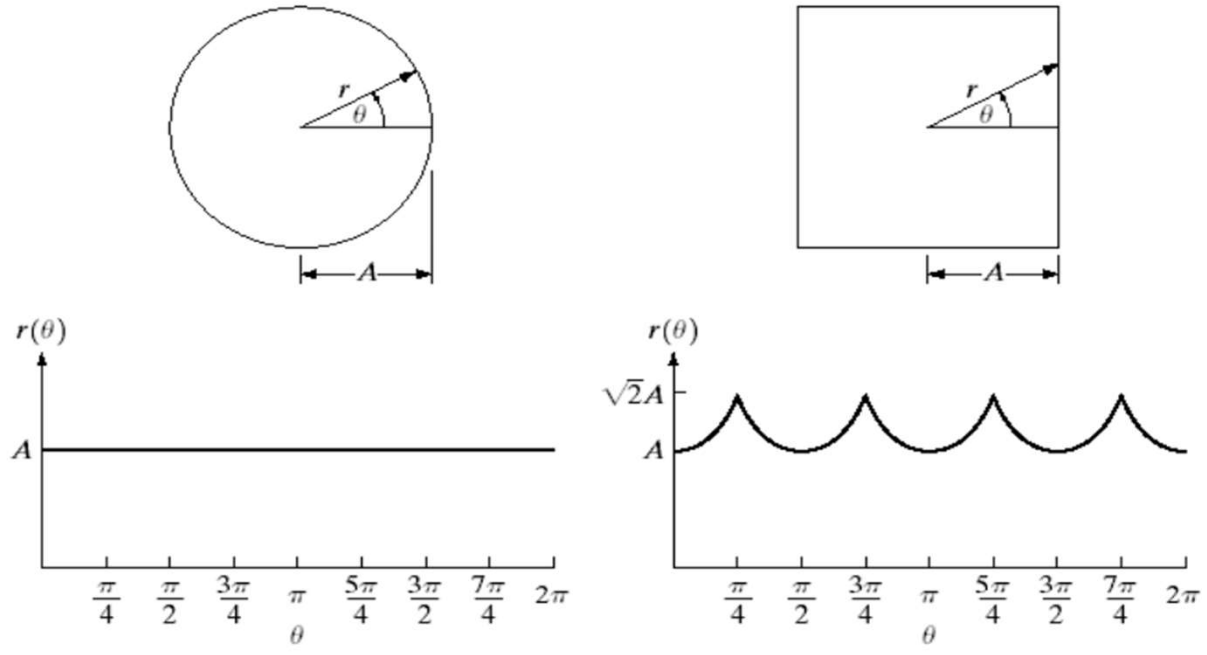


Representation: Signatures

- Signature is 1-D representation of 2-D closed curve boundary of the object
- Signature is a plot of the distance from the centroid to the boundary as a function of angle
- The signature is a unique representation for different shape of the object
  - We can use the signature to distinguish the object by it's shape
- Signature generated by this method is translation invariant but not rotation & scale invariant

Representation: Signatures-Example

Represent an 2-D object boundary in term of a 1-D function of radial distance with respect to θ.



a b

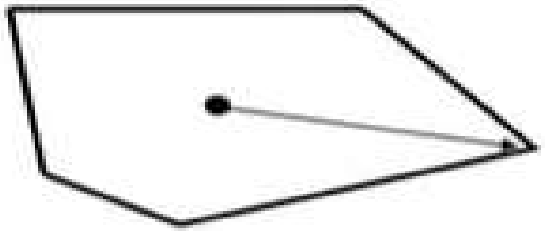
FIGURE 11.10 Distance-versus-angle signatures. In (a) $r(\theta)$ is constant. In (b), the signature consists of repetitions of the pattern $r(\theta) = A \sec\theta$ for $0 \leq \theta \leq \pi/4$ and $r(\theta) = A \csc\theta$ for $\pi/4 < \theta \leq \pi/2$.

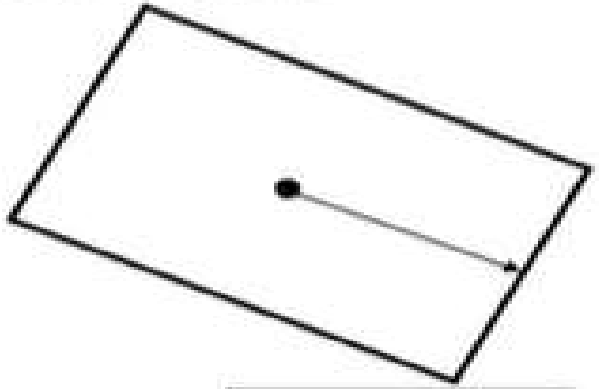
Signature is translation invariant

- as we move this square somewhere else signature remains the same because signature is w.r.t centroid

Signature is not rotation invariant

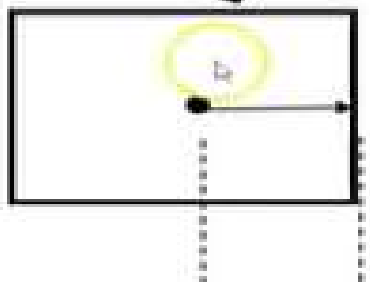
- for square or rectangle it is rotation invariant because as we rotate the square it's nature remains same
- But it is not invariant w.r.t starting point
- To make it invariant w.r.t starting point, we will define the starting point with some criteria
- Criteria is starting point is the farthest point from centroid
- In case of square 4 corners can be starting point to make it ***rotation invariant***





Signature is not scale invariant

- Let we reduce the size of rectangle,
  - Nature of signature remains same but amplitude will change
- To make signature invariant w.r.t size or scale, we normalise the amplitude in the range [0 1] , then the signature becomes ***scale invariant***



Normalization for rotation

(1) Choose the starting point as the furthest point from the centroid OR

(2) Choose the starting point as the point on the major axis that is the furthest from the centroid

Normalization for scale Note: $\uparrow$ scale $\Rightarrow$ $\uparrow$ amplitude of signature

(1) Scale signature between 0 and 1 Problem: sensitive to noise

(2) Divide each sample by its variance - assuming it is not zero

Alternative approach: plot $\Phi(\theta)$

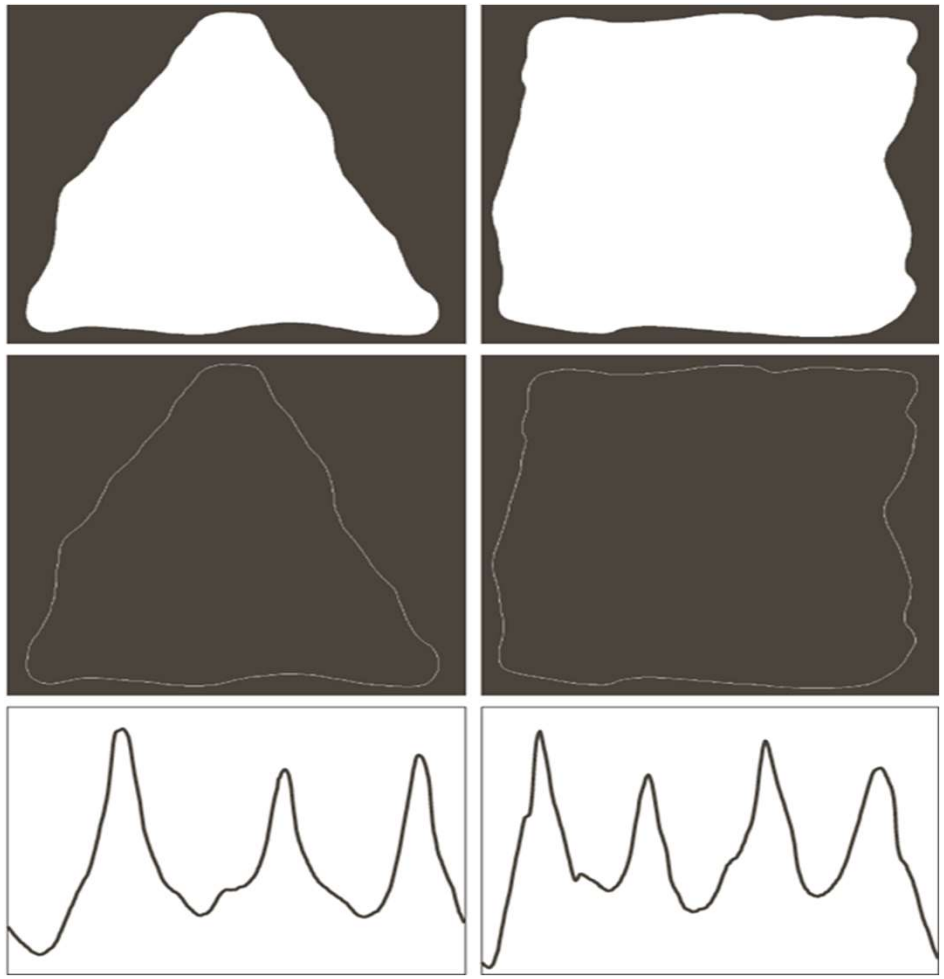
- Φ: angle between the line tangent to the boundary and a reference line
- θ: angle with the positive x-axis

$\Phi(\theta)$ carry information about basic shape characteristics

Alternative approach: use the so-called slope density function as a signature, that is a histogram of the tangent-angle values

- Respond strongly to sections of the boundary with constant tangent angles (straight or nearly straight segments)
- Deep valleys in sections producing rapidly varying angles (corners or other sharp inflections)

Representation: Signatures-Example



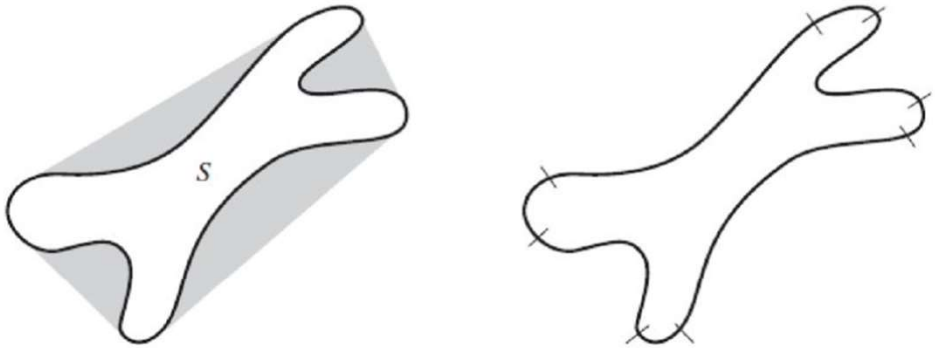
a b
c d
e f

FIGURE 11.11
Two binary regions, their external boundaries, and their corresponding $r(\theta)$ signatures. The horizontal axes in (e) and (f) correspond to angles from 0° to 360°, in increments of 1°.

Representation –Boundary Segments

- Boundary Segments
  - Decomposing a boundary into segments is often useful.
  - Decomposition reduces the boundary's complexity and thus simplifies the description process.
  - In this case, use of the convex hull of the region enclosed by the boundary is a powerful tool for robust decomposition of the boundary.

- Boundary Segments
  - The *convex hull H* of an arbitrary set *S* is the smallest convex set containing *S*.
  - The set difference *H* – *S* is called the *convex deficiency D* of the set *S*.
  - Figure 11.12(b) shows the result in this case.
  - Note that, in principle, this scheme is independent of region size and orientation.



a b

Figure 11.12
(a) A region, and its convex deficiency (shaded).
(b) Partitioned boundary.

Representation : Skeleton- Medial Axis Transformation

Skeletonization, that is the process that reduces the boundary of a region to a graph (which is a single pixel thick), often precedes other representation schemes and is therefore discussed first..

Brute force method: Medial axis transformation (MAT - 1967)

Consider a region R with boundary B...

- For each point p in R find its closest neighbour in B
- If p has more than one closest neighbour, then p is part of the medial axis

- **MAT algorithm:**
  (1) for each point in the region we find its closest point in boundary,
  (2) if a point has more than one such a neighbor --> a point belongs to the medial axis (skeleton) of the region
- the results of MAT operation depend on the distance measure:

pixel	coordinates
p	(x,y)
q	(s,t)

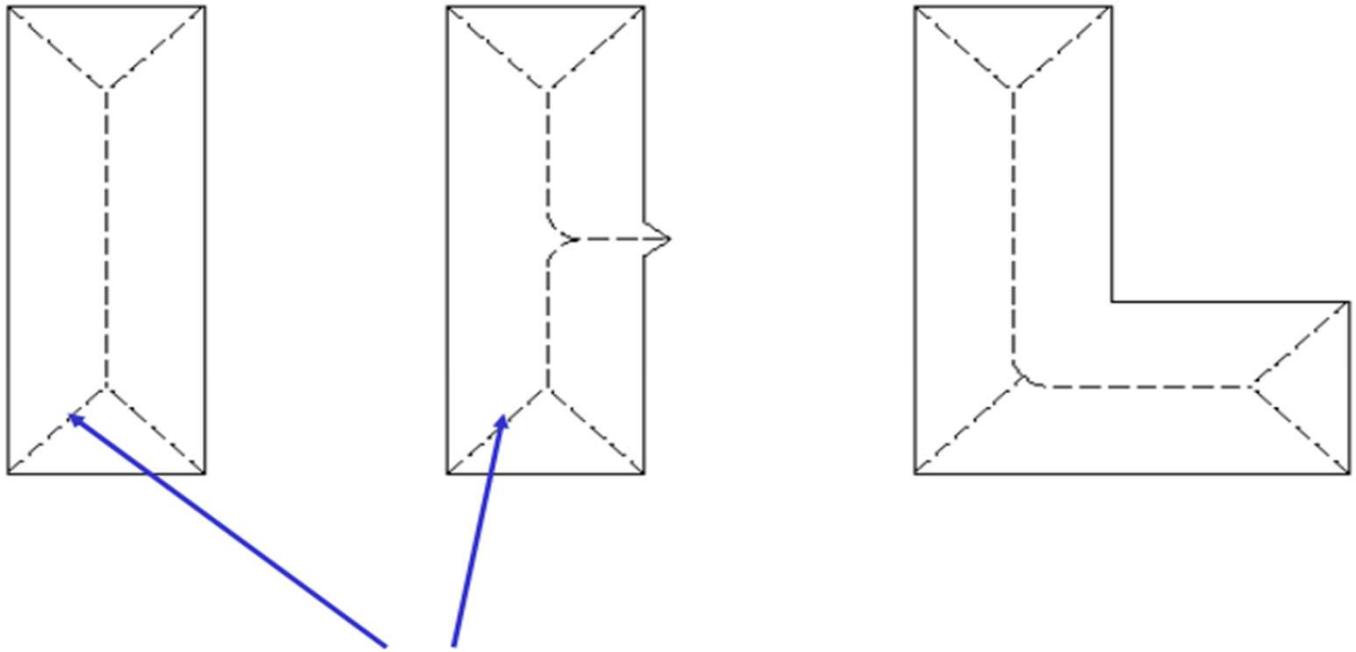
(a) **Euclidean distance** between p and q is defined as: $D_e(p,q) = [(x- s)^2 + (y - t)^2]^{1/2}$

(b) **D_4 distance (City block distance)** : $D_4(p,q) = | x- s | + | y - t |$

(c) **D_8 distance(Chessboard distance)**: $D_8(p,q) = \max(| x- s |, | y - t |)$

- Direct implementation of MAT is computationally expensive.
- Alternative algorithms have been proposed that "thin" the boundary of a region until the skeleton is left.

Obtained from thinning or skeletonizing processes



a b c

FIGURE 11.7
Medial axes (dashed) of three simple regions.

Representation : Skeleton- Thinning Algorithm

Thinning algorithm:

Edge points are deleted in an iterative way so that

(1) end points are not removed, (2) connectivity is not broken, and

(3) no excessive erosion is caused to the region

This algorithm thins a binary region, where an edge point $= 1$ and a background point $= 0$

Contour point: Edge point $(= 1)$ with at least one neighbour with a value of 0

Step 1: A contour point is flagged for deletion if

(a) $2 \leq N(p_1) \leq 6$

(b) $T(p_1) = 1$

(c) $p_2 \cdot p_4 \cdot p_6 = 0$

(d) $p_4 \cdot p_6 \cdot p_8 = 0$

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

Example

0	0	1
1	p_1	0
1	0	1

$N(p_1) = 4$

$T(p_1) = 3$

$N(p_1) \equiv$ **number of non-zero neighbours of** p_1

$T(p_1) \equiv$ **number of** $0-1$ **transitions in sequence:** $\{p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_2\}$

Now delete all the flagged contour points and consider the remaining contour points...

Step 2: A contour point is flagged for deletion if

(a) $2 \leq N(p_1) \leq 6$

(b) $T(p_1) = 1$

(c') $p_2 \cdot p_4 \cdot p_8 = 0$

(d') $p_2 \cdot p_6 \cdot p_8 = 0$

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

Delete all the flagged contour points; Repeat steps 1 and step 2 until no contour point is deleted during an iteration

Reasons for each of these conditions...

(a) $N(p_1) = 1$:

0	**1**	**0**
0	**1**	**0**
0	**0**	**0**

: end point will be deleted!

$N(p_1) = 7$:

1	**1**	**1**
1	**1**	**1**
1	**0**	**1**

: erosion will occur!

(b) $T(p_1) = 2$:

0	0	1
0	1	0
1	0	0

: connectivity will be broken!

Note that $N(p_1) = 2$ **here**

Reasons for conditions (c), (d), (c') and (d'): see page 836

Example 11.5: The skeleton of a region

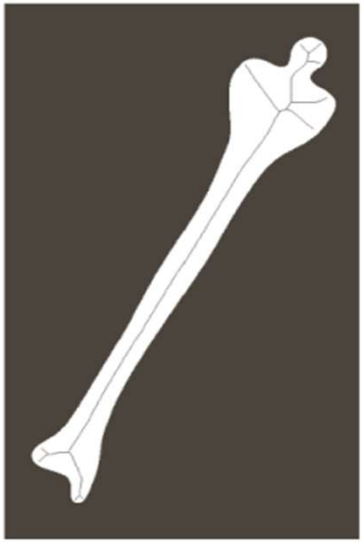


FIGURE 11.16
Human leg bone and skeleton of the region shown superimposed.

a) This means check isolated point at least 2-6 neighbours

b) It is not a corner or Junction point.

c) d) = at least any one pair is zero .

It checks it is a part of Horizontal, vertical, Diagonal line.

C'),d') = at least one is zero in the pair. It Check part of opposite diagonals.

After checking all only make it as zero.

The above are the algorithm explanation.