DIGITAL IMAGE PROCESSING

UNIT-V

Image Representation: Shape Features (Region-based representation and descriptors) Area Euler's Number Eccentricity Elongatedness Rectangularity Direction Compactness – Moments Convex Hull Texture Features Color Features. Object and Pattern Recognition: Pattern and Pattern Classes Matching,

Ω ***Image representation and description:***

This involves representing an image in a way that can be analyzed and manipulated by a computer, and describing the features of an image in a compact and meaningful way

Ω ***Shape features***

In computer vision and image processing, a **feature** is a piece of information about the content of an image; typically about whether a certain region of the image has certain properties. Features may be specific structures in the image such as points, edges or objects. Features may also be the result of a general neighborhood operation or **feature detection** applied to the image. Other examples of features are related to motion in image sequences, or to shapes defined in terms of curves or boundaries between different image regions. More broadly a *feature* is any piece of information which is relevant for solving the computational task related to a certain application. This is the same sense as feature in machine learning and pattern recognition generally, though image processing has a very sophisticated collection of features. The feature concept is very general and the choice of features in a particular computer vision system may be highly dependent on the specific problem at hand

Ω Image Representation & Description

• After an image is segmented into regions, the regions are represented and described in a form suitable for computer processing (descriptors).

• Representing a region:

1. In terms of its external characteristics (boundary)

2. In term of its internal characteristics

Exp: A region might be represented by the length of its boundary.

– External representations are used when the focus is on shape of the region.

– Internal representations are used when the focus is on pixel.

Ω *Geometrical Based Features:*

Shanmugavadivu et al [11] in the year 2012, have proposed 7 geometric based properties, 2 margin based properties for shape based image retrieval applications.

(a) Equivalent Diameter:

Equivalent Diameter (ED) is used to compute the silhouette of image diameter that the region has the same area. It is given in Eq. (1)

$$ED = \frac{2 * \sqrt{\text{Area} * \pi}}{\pi} \qquad \text{.... (1)}$$

(b) Area:

The area can be determined by the actual number of pixels in the image region. Each pixel has different weights. This value differs from the binary image area and it is given below

$$Area = \pi * (\text{Radius})^2 \quad \text{... (2)}$$

(c) Major Axis length:

Major Axis Length can be determined to the Image pixel distance between the major axis endpoint and given by the relation. This property only supported for 2-D input label matrices. The result is to measure the object length and it is

$$Major\ Axis\ Length = \sqrt{\left((index1) - x(index2)\right)^2 + \left(y(index1) - y(index2)\right)^2}$$

... (3)

(d) Roundness:

Roundness can be determined by how to close of the object to the circle shape. The perimeter value determines the length of the boundary of the object. Which is given below

$$Roundness = \frac{(4 * obj\_area * \pi)}{Perimeter^2} \quad \text{... (4)}$$

(e) Radius:

Radius can be specified in the half-length of its diameter. It is given below

$$Radius = \frac{Diameter}{2} \quad \text{... (5)}$$

(f) Compactness:

The ratio of the area of an object to the area of a circle with the same perimeter. The mean squared distance of the object's pixels from the centroid divided by the area. A filled circle will have compactness (CN) of 1, with irregular objects or objects with holes having a value greater than 1. It is given in eq (6)

$$CN = \left(\frac{2 * \sqrt{\text{Area} * \pi}}{\text{Perimeter}} \right) \quad \text{... (6)}$$

(g) Elongatedness:

Elongatedness (EN) is the Ratio between the length and width of the object bounding box. The ratio value is between o and 1. It is given below

$$EN = \left(\frac{Area}{(2 * maxRdius)^2} \right) \quad \text{... (7)}$$

(h) Eccentricity:

Eccentricity (ECT) of the image has the same normalized second central moments as a region. The range of the value between 0 and 1. Eccentricity value can be determined to the Ratio of the major axis length and minor axis length. Which is presented below

$$ECT = \sqrt{\left(1 - \left(\frac{minRadius}{maxRadius} \right) \right)^2} \quad \text{... (8)}$$

Ω ***What is a Texture?***

The natural world is rich in texture:

the surface of any visible object is textured at certain scale. A wealth of textures are observed on both artificial and natural objects such as those on wood, plants, materials and skin. In a general sense, the word texture refers to surface characteristics and appearance of an object given by the size, shape, density, arrangement, proportion of its elementary parts .

A texture is usually described as smooth or rough, soft or hard, coarse of fine, matt or glossy, and etc.

Textures might be divided into two categories,
namely**, *tactile* and *visual* textures**. Tactile textures refer to the immediate tangible feel of a surface.

Visual textures refer to the visual impression that textures produce to human observer, which are related to local spatial variations of simple stimuli like colour, orientation and intensity in an image. This thesis focuses only on visual textures, so the term `texture' thereafter is exclusively referred to `visual texture' unless mentioned otherwise.

- Although texture is an important research area in computer vision, there is no precise definition of the notion texture.
- The main reason is that natural textures often display different yet contradicting properties, such as regularity versus randomness, uniformity versus distortion, which can hardly be described in a unified manner.
- Haralick considers a texture as an ``organised area phenomenon'' which can be decomposed into `primitives' having specific spatial distributions .This definition, also known as *structural approach*, comes directly from human visual experience of textures. For instance,
- a texture is ``a stochastic, possibly periodic, two-dimensional image field'' . This definition describes a texture by a stochastic process that generates the texture, which is also known as *stochastic approach*. These different

definitions usually lead to different computational approaches to texture analysis.

Figure 4.1: Examples of natural textures.

Figure 4.2: Examples of artificial regular textures.

Nevertheless, an apparent consensus that spatial homogeneity is the most important property of a texture has been reached. the patterns at different magnifications, although not identical, are represented by the same signal statistics

Textures also exhibit local non-homogeneity, i.e. departures from strict homogeneity to some extent in a local image region. For example, in the image `leaves' every single leaf is slightly different from another (local non-homogeneity), but as a whole they display approximate spatial uniformity and consistency (global homogeneity).

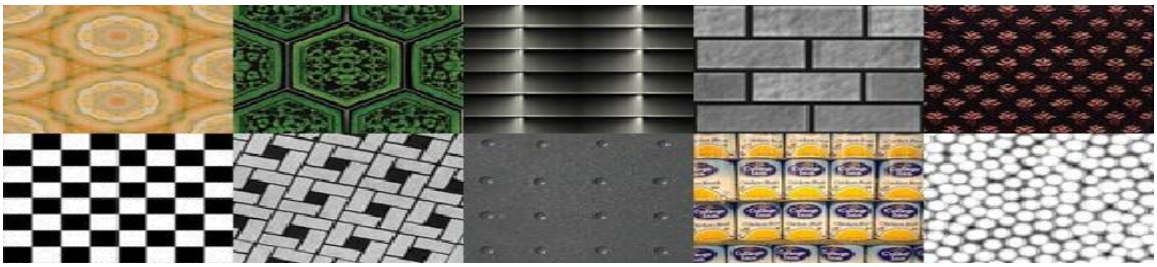


Figure 4.3: Examples of natural regular textures.

Due to the diversity and complexity of natural textures, it is useful to separate them into categories. For instance, **textures can be classified into regular and stochastic ones by their degree of randomness**. A *regular* texture is formed by regular tiling of easily identifiable small size elements organised into strong periodic patterns.

A *stochastic* texture exhibits less noticeable elements and display rather random patterns. For examples, Most of real world textures, however, are mixtures of the above-mentioned categories.

Figure 4.4: Examples of stochastic textures.

By spatial homogeneity, **textures can be classified into *homogeneous*, *weakly-homogeneous*, and *inhomogeneous* patterns. Specifically, homogeneous texture** contains ideal repetitive structures, and such uniformity produces idealised patterns. Weak homogeneity involves local spatial variation in texture elements or their spatial arrangement,

which leads to more or less violates the precise repetitiveness An `inhomogeneous texture' mostly refers to an image where repetition and spatial self-similarity are absent.

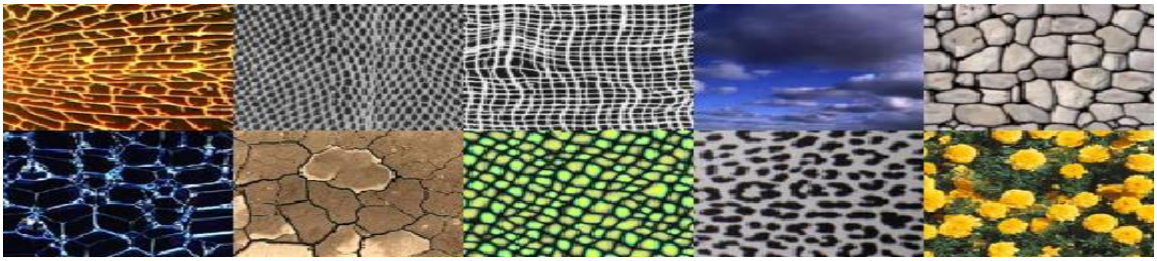


Figure 4.5: Examples of weakly-homogeneous textures.

Ω ***COLOR FEATURE EXTRACTION***

- Colour space represents the color in the form of intensity value. We can specify, visualize and create the color by using color space method.
- There are different color feature extraction methods. Color feature extraction methods
- **Histogram Intersection Method:** Histogram Intersection (HI) considers global color features. The duos of color histograms X and Y with k bins for each, HI is defined as, In Histogram Intersection method, the number of bins makes impact on performance. The large no of bins represent the image in very complex manner it increases the computational complexity.
- **Zernike Chromaticity Distribution Moments**: It is derived from chromaticity space. This method gives fixed length and computation effective representation of an image which contains the color content of an image but, their size is invariant under rotation and flipping.
- **Color Histogram:** Color histogram represents the image from different perspective. The image in which color bins of frequency distribution are represented by color histogram and it counts the pixels which are similar and store it. Color histogram analyzes every statistical color frequency in an image. The change occurred in the translation, rotation and angle of view these problems are solved by color histogram and also it focuses on

individual parts of an image. The computation of local color histogram is easy and it is resistant to minor variations in the image so for indexing and retrieval of image database it is very important.

❖Object and Pattern Recognition

Introduction:

pattern is everything around in this digital world. A pattern can either be seen physically or it can be observed mathematically by applying algorithms.

Example: The colors on the clothes, speech pattern, etc. In computer science, a pattern is represented using vector feature values.

What is Pattern Recognition?

Pattern recognition is the process of recognizing patterns by using a machine learning algorithm. Pattern recognition can be defined as the classification of data based on knowledge already gained or on statistical information extracted from patterns and/or their representation. One of the important aspects of pattern recognition is its application potential.

Examples: Speech recognition, speaker identification, multimedia document recognition (MDR), automatic medical diagnosis.

In a typical pattern recognition application, the raw data is processed and converted into a form that is amenable for a machine to use. Pattern recognition involves the classification and cluster of patterns.

- In classification, an appropriate class label is assigned to a pattern based on an abstraction that is generated using a set of training patterns or domain knowledge. Classification is used in supervised learning.
- Clustering generated a partition of the data which helps decision making, the specific decision-making activity of interest to us. Clustering is used in unsupervised learning.

Features may be represented as continuous, discrete, or discrete binary variables. A feature is a function of one or more measurements, computed so that it quantifies some significant characteristics of the object.

Example: consider our face then eyes, ears, nose, etc are features of the face. A set of features that are taken together, forms the **features vector**.
Example: In the above example of a face, if all the features (eyes, ears, nose, etc) are taken together then the sequence is a feature vector([eyes, ears, nose]). The feature vector is the sequence of a feature represented as a d-dimensional column vector. In the case of speech, MFCC (Mel-frequency Cepstral Coefficient) is the spectral feature of the speech. The sequence of the first 13 features forms a feature vector.

Pattern recognition possesses the following features:

- Pattern recognition system should recognize familiar patterns quickly and accurate
- Recognize and classify unfamiliar objects
- Accurately recognize shapes and objects from different angles
- Identify patterns and objects even when partly hidden
- Recognize patterns quickly with ease, and with automaticity.

Advantages:

- Pattern recognition solves classification problems
- Pattern recognition solves the problem of fake biometric detection.
- It is useful for cloth pattern recognition for visually impaired blind people.
- It helps in speaker diarization.
- We can recognize particular objects from different angles.

Disadvantages:

- The syntactic pattern recognition approach is complex to implement and it is a very slow process.
- Sometimes to get better accuracy, a larger dataset is required.
- It cannot explain why a particular object is recognized.
  Example: my face vs my friend's face.

❖ ***Applications:***

- **Image processing, segmentation, and analysis**
  Pattern recognition is used to give human recognition intelligence to machines that are required in image processing.

- **Computer vision**
  Pattern recognition is used to extract meaningful features from given image/video samples and is used in computer vision for various applications like biological and biomedical imaging.
- **Seismic analysis**
  The pattern recognition approach is used for the discovery, imaging, and interpretation of temporal patterns in seismic array recordings. Statistical pattern recognition is implemented and used in different types of seismic analysis models.
- **Radar signal classification/analysis**
  Pattern recognition and signal processing methods are used in various applications of radar signal classifications like AP mine detection and identification.
- **Speech recognition**
  The greatest success in speech recognition has been obtained using pattern recognition paradigms. It is used in various algorithms of speech recognition which tries to avoid the problems of using a phoneme level of description and treats larger units such as words as pattern
- **Fingerprint identification**
  Fingerprint recognition technology is a dominant technology in the biometric market. A number of recognition methods have been used to perform fingerprint matching out of which pattern recognition approaches are widely used.

❖ Patterns and Pattern Classes

A **pattern is an arrangement of descriptors**

The name *feature is* used often in the pattern recognition literature to denote a descriptor. A *pattern class is a family* of patterns that share some common properties. Pattern classes are denoted ω1,ω2,…….,ωW, where W is the number of classes. Pattern recognition by machine involves techniques for assigning patterns to their respective classes— automatically and with as little human intervention as possible. Three common pattern arrangements used in

practice are vectors (for quantitative descriptions) and strings and trees (for structural descriptions). Pattern vectors are

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$$

represented by bold lowercase letters, such as **x, y**, and **z**, and take the form where each component, x_i, represents the ith descriptor and n is the total number of such descriptors associated with the pattern. Pattern vectors are represented as columns (that is, n x 1 matrices). Hence a pattern vector can be expressed in the formshown in Eqn. (1) or in the equivalent form x = (x1, x2, xn)T, where ***T*** indicates transposition. The nature of the components of a pattern vector x depends on the approach used to describe the physical pattern itself. Let us illustrate with an examplethat is both simple and gives a sense of history in the area of classification of measurements. In our present terminology, each flower is described by two measurements, which leads to a 2-D pattern vector of the form

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

where x_1 and x_2 correspond to petal length and width, respectively. The three pattern classes in this case, denoted ω1, ω2, and ω3, correspond to the varieties *setosa, virginica,* and *versicolor, r*espectively. Because the petals of flowers vary in width and length, the pattern vectors describing these flowers also will vary, not only between different classes, but also within a class. The above Figure shows length andwidth measurements for several samples of each type of iris. After a set of measurements has been selected (two in this case), the components of a pattern vector become the entire description of each physical sample. Thus each flower inthis case becomes a point in 2-D Euclidean space. We note also that measurements of petal width and length in this case adequately separated the class of *Iris setosa* from the other two but did not separate as successfully the *virginica* and *versicolor* types from each other. This result illustrates the classic *feature selection* problem, in which the degree of class seperability depends strongly on the choice of descriptors selectedfor an application.

❖ RECOGINITON BASED ON MATCHING:

Recognition techniques based on matching represent each class by a prototype pattern vector. An unknown pattern is assigned to the class to which it is closest in terms of a predefined metric. The simplest approach is the minimum distance classifier, which, as its name implies, computes the (Euclidean) distance between the unknown and each of the prototype vectors. It chooses the smallest distance to makea decision. We also discuss an approach based on correlation, which can be formulated directly in terms of images and is quite intuitive.

❖ MINIMUM DISTANCE CLASSIFIER

Suppose that we define the prototype of each pattern class to be the mean vector of

$$m_j = \frac{1}{N_j}\sum_{x\epsilon\omega_j} x_j \qquad j = 1,2,\ldots\ldots\ldots\ldots,W$$

the patterns of that class:

where *NJ* is the number of pattern vectors from class ωj and the summation is taken over these vectors. As before, W is the number of pattern classes. One way to determine the class membership of an unknown pattern vector x is to assign it to the class of its closest prototype, as noted previously. Using the Euclidean distance to determine closeness reduces the problem to computing the distance measures.

$$d_{ij}\ (x) = d_i\ (x)\ -d_j\ (x)$$

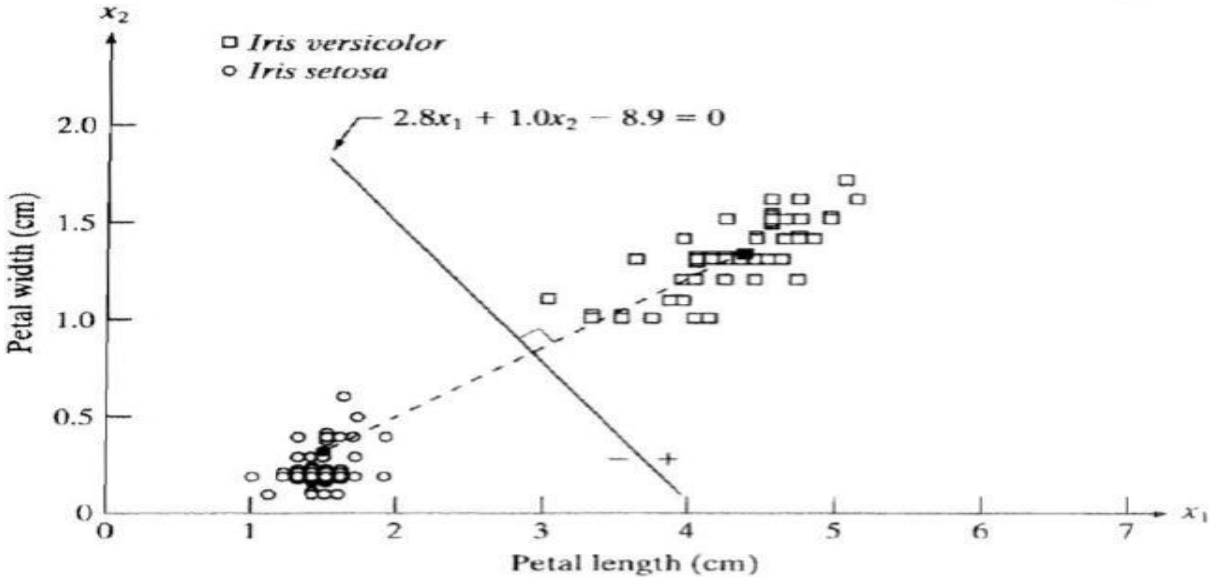
The decision boundary between classes ωi and ωj for a minimum distance classifier is

$$=x^T(m_i - m_j) - \frac{1}{2}(m_i - m_j)^T(m_i + m_j) = 0$$

The surface given by the above equation is is the perpendicular bisector of the line segment joining mmi and mmj. For n = 2, the perpendicular bi-sector is a line, for n = 3 it is a plane, and for n > 3 it is called a hyper plane. The two classes, Iris versicolor andIris setosa, denoted ω1 and ω2, respectively, have sample mean vectors m1 = $(4.3, 1.3)^T$ and m2 = $(1.5, 0.3)^T$. The decision functions are

$$d_1\ (x) = x^T m_1 - \frac{1}{2} m_1^T m_1$$

$$= 4.3\ x_1 + 1.3\ x_2 - 10.1$$



The above figure shows a plot of this boundary (note that the axes are not to the samescale). Substitution of any pattern vector from class ω1 would yield d12(x) > 0.

- Conversely, any pattern from class ω2would yield d12(x) < 0.
- In other words, give n an unknown pattern belonging to one of these two classes, the sign of d12(x) to one of these two classes, the sign of d12(x) would be sufficient to determine the pattern's class membership.

❖ MATCHING BY CORRELATION

Correlation is a mathematical technique to see how close two things are related. In image processing terms, it is used to compute the response of a mask on an image**. A mask is applied on a matrix from left to right. Mask slides over the matrix from left to right by one unit every time. Once the mask reaches the rightmost end, the mask is slid downward by one unit and again starts from left to right side.** The computed output is assigned to the central pixel, while neighbourhood pixels are also get used in the computation. The mask or the matrix can be 1-D or 2-D. Generally, the mask's dimension is taken as an odd number, so that the central pixel can easily be found.

Template matching is a technique for finding areas of an image that are similar to a patch (template). A patch is a small image with certain features. The goal of template matching is to find the patch/template in an image. To find it, the user has to give two input images: **Source Image (S)** – The image to find the template in, and **Template Image (T)** – The image that is to be found in the source image.

- It is basically a method for searching and finding the location of a template image in a larger image.
- The idea here is to find identical regions of an image that match a template we provide, giving a threshold.
- The threshold depends on the accuracy with which we want to detect the template in the source image.
- For instance, if we are applying face recognition and we want to detect the eyes of a person, we can provide a random image of an eye as the template and search for the source (the face of a person).
- In this case, since “eyes” show a large number of variations from person to person, even if we set the threshold as 50% (0.5), the eye will be detected.
- In cases where almost identical templates are to be searched, the threshold should be set high. (t>=0.8)

How does Template Matching Work?

- The template image simply slides over the input image (as in 2D convolution)
- The template and patch of input image under the template image are compared.
- The result obtained is compared with the threshold.
- If the result is greater than the threshold, the portion will be marked as detected.
- In the function cv2.matchTemplate(img_gray, template, cv2.TM_CCOEFF_NORMED) the first parameter is the main image, the second parameter is the template to be matched and the third parameter is the method used for matching.

❖ ***Optimum statistical classifiers***

Naïve Bayes Algorithm

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other. Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. All naive bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. Naive Bayes algorithm is a fast, highly scalable algorithm, which can be used for binary and multi-class classification. It depends on doing a bunch of counts. It is a popular choice for text classification, spam email classification, etc. It can be easily trained on small dataset. It has limitation as it considers all the features to be unrelated, so it cannot learn the relationship between features. Naive Bayes can learn individual features importance but can't determine the relationship among features. Different types of naïve bayes algorithms are gaussian naïve bayes, multinomial naïve bayes, and Bernoulli naïve bayes.

Previously, we had considered very simple classifiers without using any high order statistics of the underlying patterns

• If we have knowledge about some sort of statistics of the patterns, we can use information to design a classified that would minimize the probability of incorrect classifications

• Let us define the probability that the pattern x belongs to ωi as p(ωi /x)

• If our classifier incorrectly decides on ωj , then we define this error as Lij

• Average Using the Bayes' rule we obtain rj (x) = 1 p(x) X W k=1 Lkjp(x/ωk)P(ωk) •

We use this average error as our cost function, and choose the class that results in the minimum error

• All rj 's are calculated for j = 1, 2, . . . , W and the class with minimum error is selected

• Now let us select a value for Lij , a reasonable choice is to make it zero when i = j and one otherwise

• Then we have dj (x) = rj (x) = p(x/ωj)P(ωj)ging this error for all classes we obtain the average error as rj (x) = X W k=1 Lkjp(ωk /x)

To be able to calculate this optimum classifier we need to know the probability of each class (P(ω)) and the pdf of the individual patters must be known

• It is a reasonable assumption that we can infer the knowledge of P(ω) based on the problem at hand

• However, it is not an easy task to estimate p(x/ωj)

• Therefore, instead of estimating these directly, we use a parametric model and estimate a few parameters hence obtain an approximate of p(x/ωj).

❖ ***NEURAL NETWORKS CLASSIFIER***

Inspired by the properties of biological neural networks, Artificial Neural Networks are statistical learning algorithms and are used for a variety of tasks, from relatively simple classification tasks to computer vision and speech recognition.

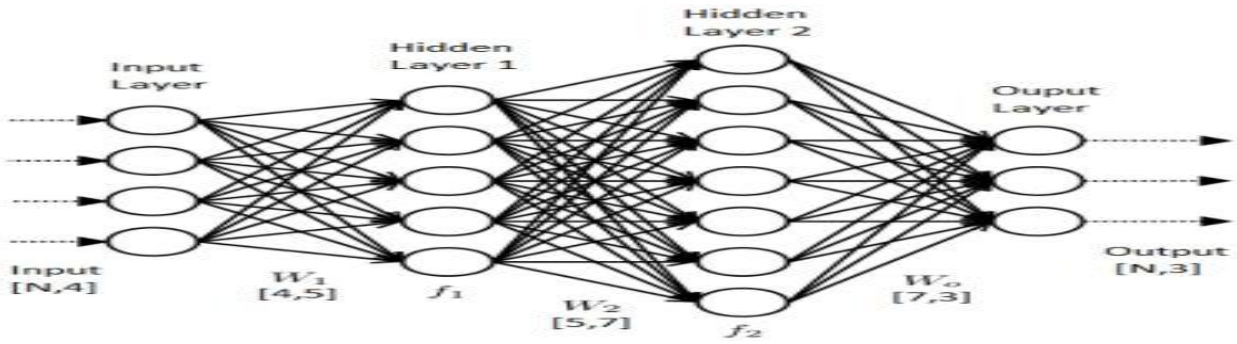
Artificial neural networks are implemented as a system of interconnected processing elements, called nodes, which are functionally analogous to biological neurons.

The connections between different nodes have numerical values, called weights, and by altering these values in a systematic way, the network is eventually able to approximate the desired function.

The hidden layers can be thought of as individual feature detectors, recognizing more and more complex patterns in the data as it is propagated throughout the network. For example, if the network is given a task to recognize a face, the first hidden layer might act as a line detector, the second hidden takes these lines as input and puts them together to

form a nose, the third hidden layer takes the nose and matches it with an eye and so on, until finally the whole face is constructed.

This hierarchy enables the network to eventually recognize very complex objects. The different types of artificial neural network are convolution neural network, feed forward neural network, probabilistic neural network, time delay neural network, deep stacking network, radial basis function network, and recurrent neural network.

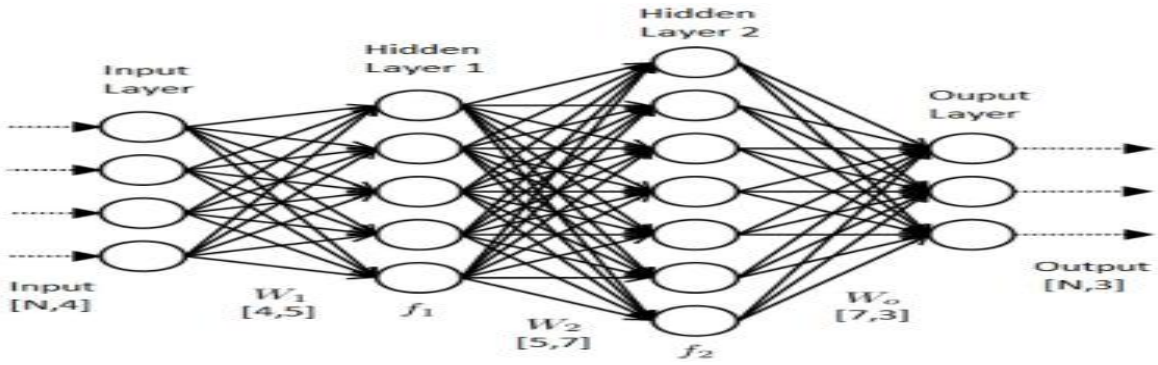


Deep neural networks: the “how” behind image recognition and other computer vision techniques

- Image recognition is one of the tasks in which **deep neural networks** (DNNs) excel. Neural networks are computing systems designed to recognize patterns.
- Their architecture is inspired by the human brain structure, hence the name.
- They consist of three types of layers: input, hidden layers, and output. The input layer receives a signal, the hidden layer processes it, and the output layer makes a decision or a forecast about the input data. Each network layer consists of interconnected *nodes (artificial neurons*) that do the computation.

What makes a neural network deep?

The number of hidden layers: While traditional neural networks have up to three hidden layers, deep networks may contain hundreds of them.



How neural networks learn to recognize patterns

How do we understand whether a person passing by on the street is an acquaintance or a stranger (complications like short-sightedness aren't included)? We look at them, subconsciously analyze their appearance, and if some inherent features – face shape, eye color, hairstyle, body type, gait, or even fashion choices – match with a specific person we know, we recognize this individual. This brainwork takes just a moment.

So, to be able to recognize faces, a system must learn their features first. It must be trained to predict whether an object is X or Z. Deep learning models learn these characteristics in a different way from machine learning (ML) models. That's why model training approaches are different as well.

Training deep learning models (such as neural networks)

To build an ML model that can, for instance, predict customer churn, data scientists must specify what input features (problem properties) the model will consider in predicting a result. That may be a customer's education, income, lifecycle stage, product features, or modules used, number of interactions with customer support and their outcomes. The process of constructing features using domain knowledge is called *feature engineering.*

If we were to train a deep learning model to see the difference between a dog and a cat using feature engineering… Well, imagine gathering characteristics of billions of cats and dogs that live on this planet. We can't construct accurate features that will work for each possible image

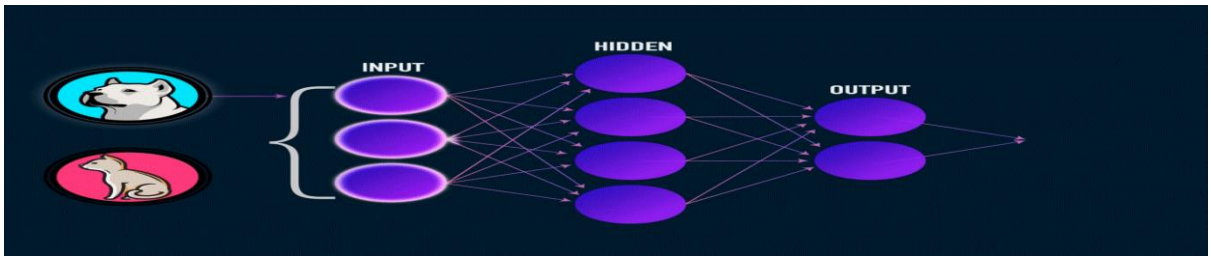
while considering such complications as viewpoint-dependent object variability, background clutter, lighting conditions, or image deformation. There should be another approach, and it exists thanks to the nature of neural networks.

Neural networks learn features directly from data with which they are trained, so specialists don't need to extract features manually.

"The power of neural networks comes from their ability to learn the representation in your training data and how to best relate it to the output variable that you want to predict. In this sense, neural networks learn mapping. Mathematically, they are capable of learning any mapping function and have been proven to be universal approximation algorithms," notes Jason Brownlee in *Crash Course On Multi-Layer Perceptron Neural Networks*.

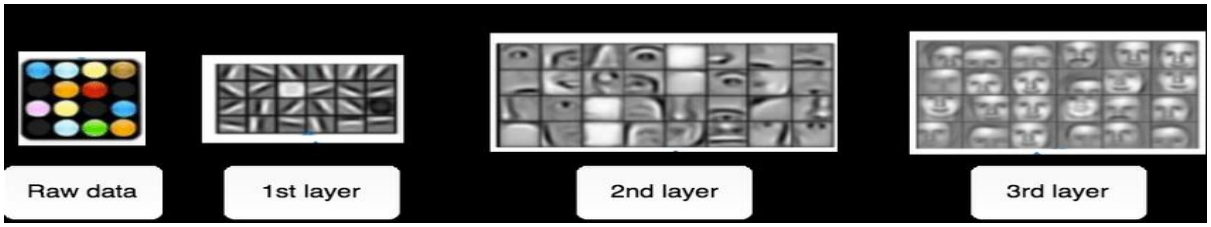
The training data, in this case, is a large dataset that contains many examples of each image class. When we say a large dataset, we really mean it. For instance, the ImageNet dataset contains more than 14 million human-annotated images representing 21,841 concepts (synonym sets or synsets according to the WordNet hierarchy), with 1,000 images per concept on average.

Each image is annotated (labeled) with a category it belongs to – a cat or dog. The algorithm explores these examples, learns about the visual characteristics of each category, and eventually learns how to recognize each image class. This model training style is called *supervised learning*.



each layer of nodes trains on the output (feature set) produced by the previous layer. So, nodes in each successive layer can recognize more complex, detailed features – visual representations of what the image

depicts. Such a “hierarchy of increasing complexity and abstraction” is known as ***feature hierarchy***.



So, the more layers the network has, the greater its predictive capability.