UNIT–5 : Image Pattern Classification (PYQ Solution)

AKTU · Image Analytics (BCDS061) · Solutions by **Chirag Jain**, Ycotes

SECTION A Answers (≈70 Words Each)

Q1. Full form of SIFT and define it. (Jun 2023)

SIFT stands for Scale-Invariant Feature Transform. It is an algorithm that detects and describes local keypoints in an image that stay the same under changes of scale, rotation and (partly) illumination and viewpoint. Keypoints are found as extrema in a difference-of-Gaussians scale space, and each is described by a 128-element gradient-orientation histogram vector.

Q2. What do you mean by pattern classes? (Jun 2023)

A pattern is an arrangement of descriptors (features), written as a vector $x = (x_1, \ldots, x_n)$. A pattern class is a family of patterns that share some common properties, denoted $\omega_1, \omega_2, \ldots, \omega_W$. Pattern recognition assigns an unknown pattern to its correct class. Example: classes of flowers such as setosa and versicolor, described by petal length and width.

Q3. Define Optimum (Bayes) statistical classifiers. (Jun 2023)

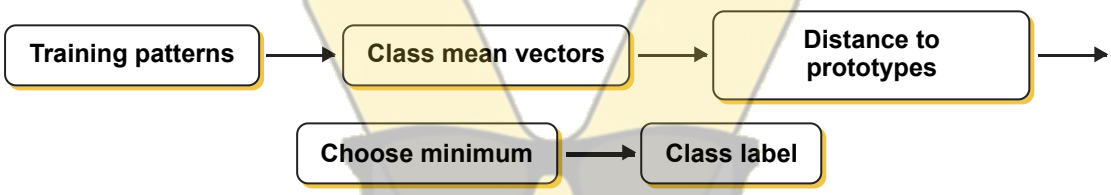
A Bayes classifier assigns a pattern x to the class ω_i with the highest posterior probability, which minimises the average probability of classification error. Using Bayes' rule, the decision function is $d_j(x) = p(x \mid \omega_j)\, P(\omega_j)$; x belongs to ω_i if $d_i(x) > d_j(x)$ for all $j \neq i$. It needs the class-conditional densities and priors, often assumed Gaussian.

SECTION B Answers (≈250 Words Each)

Q1. Define patterns. Also discuss pattern classification by prototype matching. (Jun 2023)

A **pattern** is an arrangement of descriptors (features) that characterise an object. It is represented as a **feature vector** $x = (x_1, x_2, \ldots, x_n)^T$ in n-dimensional space (quantitative), or as strings and trees (structural). A set of patterns with common properties forms a **pattern class** ω_i. **Pattern classification** assigns an unknown pattern to one of the known classes.

Classification by prototype matching represents each class by a **prototype** (typically the **mean vector**) and assigns the unknown pattern to the class whose prototype is nearest.



Prototype (minimum distance) classifier

Minimum distance classifier

1. For each class j compute the prototype $m_j = (1/N_j) \Sigma x$ over patterns of class j.
2. For an unknown x compute the Euclidean distance $D_j(x) = \|x - m_j\|$.
3. Assign x to the class with the smallest D_j.

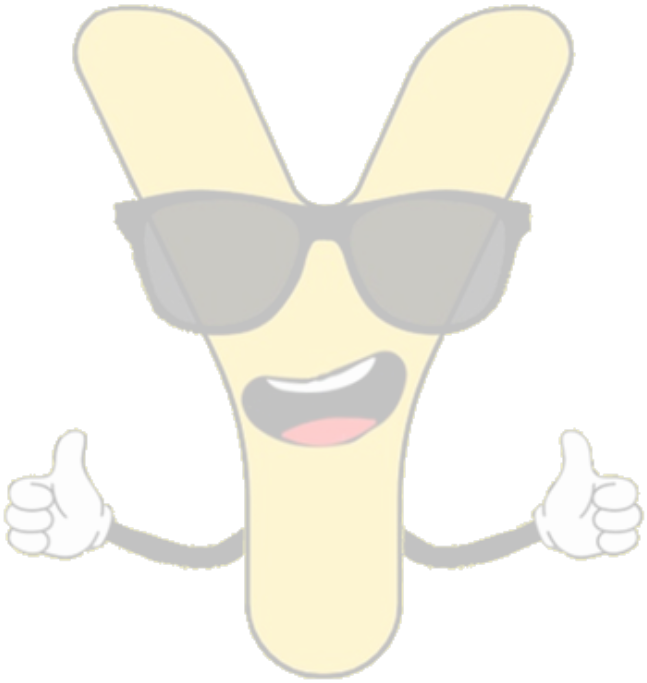
Equivalent decision function: $d_j(x) = x^T m_j - \tfrac{1}{2} m_j^T m_j$; choose the class with the largest value. The decision boundary between two classes is the **perpendicular bisector** of the line joining their means.

Example: $m_1 = (4, 1.5)$, $m_2 = (1.5, 0.3)$; unknown $x = (3.5, 1.2)$. $D_1 = \sqrt{(0.25 + 0.09)} = 0.58$; $D_2 = \sqrt{(4 + 0.81)} = 2.19$. So x is assigned to class 1.

Other forms: correlation matching, where the pattern is correlated with a template image and the class with maximum normalised correlation is chosen: $\gamma(x, y) = \Sigma w(s, t) f(x+s, y+t)$.

Advantage	Limitation
Simple, fast	Poor if classes overlap
Few parameters	Assumes compact, spherical classes
Works with small data	Sensitive to scale of features

In conclusion, prototype matching is a simple, effective classifier when classes are well separated.

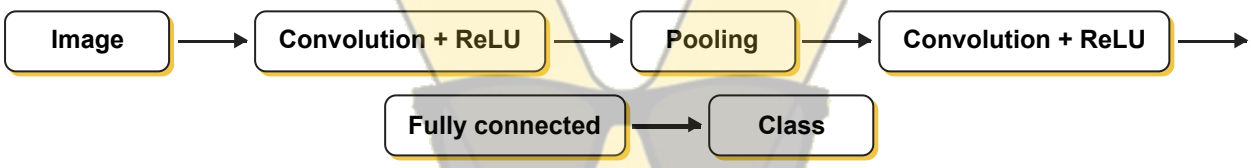


SECTION C Answers (≈500 Words Each)

Q1. Differentiate between multilayer feed-forward neural networks and deep convolutional neural networks in detail. (Jun 2023)

Multilayer feed-forward network (MLP): neurons are arranged in an input layer, one or few hidden layers and an output layer; each neuron is **fully connected** to all neurons of the next layer, and signals move forward only. Each unit computes $a = f(\Sigma w_i x_i + b)$. Training uses backpropagation. The input is a flattened feature vector.

Deep convolutional neural network (CNN): a deep network designed for images, made of **convolutional layers** (small shared filters that slide over the image to make feature maps), **activation (ReLU)**, **pooling layers** (down-sampling), and finally fully connected layers with softmax. It learns features directly from raw pixels, from edges in early layers to objects in deeper layers.



Structure of a deep CNN

Point	Multilayer feed-forward	Deep CNN
Connectivity	Fully connected	Local receptive fields
Weights	Separate for every connection	**Shared** filter weights
Parameters	Very large for images	Far fewer
Input	Hand-crafted feature vector	Raw image (2-D / 3-D)
Feature extraction	Done separately by designer	Learned automatically
Spatial structure	Lost on flattening	Preserved
Invariance	Not translation invariant	Tolerant to shifts via pooling
Depth	Few layers	Many layers (tens or hundreds)
Layer types	Dense only	Convolution, pooling, dense, normalisation
Data and compute	Smaller	Large data, GPUs
Examples	Simple classifier	LeNet, AlexNet, VGG, ResNet

Example: for a 100 × 100 image and 1000 hidden neurons an MLP needs 10^7 weights in the first layer; a convolutional layer with 32 filters of 3 × 3 needs only 320.

In conclusion, CNNs exploit image structure and give better accuracy with fewer parameters, whereas MLPs suit small feature-based problems.

Q2. Write short notes on: (i) Neural networks, (ii) Deep learning. (Jun 2023)

(i) Neural networks

An **artificial neural network (ANN)** is a computing model inspired by the brain, made of interconnected **neurons** (perceptrons).

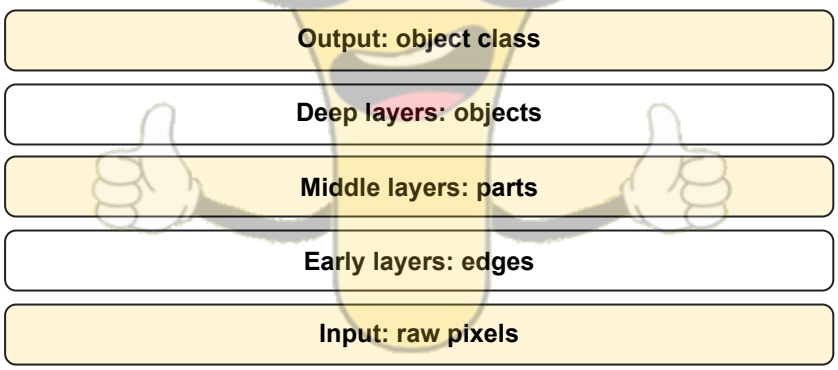
- **Neuron:** output $y = f(\Sigma\, w_i x_i + b)$, with weights w, bias b and activation function f (step, sigmoid, tanh, ReLU).
- **Architecture:** input layer, hidden layers, output layer; feed-forward or recurrent.
- **Learning:** weights are adjusted to minimise a loss (e.g. squared error). The **backpropagation** algorithm computes gradients layer by layer and updates $w \leftarrow w - \eta\, \partial E/\partial w$ (gradient descent, learning rate η).

- **Perceptron:** single neuron; linearly separable problems only. Multilayer networks solve non-linear problems (XOR).
- **Uses in imaging:** character recognition, classification, segmentation, face recognition.
- **Issues:** overfitting, local minima, need of training data.

(ii) Deep learning

Deep learning is a branch of machine learning that uses neural networks with **many hidden layers** to learn **hierarchical representations** from raw data automatically.

- Early layers learn simple features (edges), middle layers parts, deep layers whole objects.
- **Architectures:** CNN (images), RNN/LSTM (sequences), autoencoders, GANs, transformers.
- **Enablers:** big data, GPUs, ReLU, dropout, batch normalisation, better optimisers (Adam).
- **Advantages:** no manual feature engineering, high accuracy.
- **Limitations:** needs large data and compute, hard to interpret.
- **Applications:** image classification, object detection, medical diagnosis, speech and language.



Hierarchical features in deep learning
