

UNIT 6

Expert Systems and Machine Vision

(4 Hours) — Artificial Intelligence (CACS410)

⚠ Please Note

These notes were prepared for quick revision and **may contain errors or omissions**. Please take care: cross-check important definitions, formulas, algorithms and worked examples against your prescribed textbook, class notes and teacher's instructions before relying on them (especially for exams).

6.1 Expert Systems

Introduction

An **expert system (ES)** is a computer program that emulates the decision-making ability of a **human expert** in a narrow domain by using a stored **knowledge base** of facts and rules and an **inference engine** that reasons with them. It can give advice, diagnose problems and explain its conclusions.

System	Domain	Notes
DENDRAL (1960s)	Chemistry	Determines molecular structure from mass-spectrometer data
MYCIN (1970s)	Medicine	Diagnoses blood infections and recommends antibiotics using about 600 rules and certainty factors
PROSPECTOR	Geology	Assesses the potential of mineral deposits
XCON / R1	Computer configuration	Configured DEC VAX computer orders

Characteristics

- Captures the **expertise** of one or more human experts in a **limited domain**.
- Uses **symbolic reasoning** (rules, heuristics) rather than purely numerical algorithms.
- Can handle **uncertain or incomplete** information (certainty factors, probabilities).
- Provides an **explanation** of how and why it reached a conclusion.
- Keeps **knowledge separate** from the reasoning mechanism so it can be updated easily.

Point	Conventional program	Expert system
Knowledge and processing	Mixed together in code	Knowledge base separate from inference engine
Approach	Algorithmic, step by step	Heuristic, rule-based reasoning
Output	Always a definite answer	Can give reasoned advice with confidence
Explanation	Usually none	Explains “how” and “why”
Change	Reprogramming needed	Add or edit rules

Advantages: available 24×7, consistent, preserves scarce expertise, faster than a human, reduces cost and errors. **Limitations:** narrow domain, no common sense, cannot learn by itself, difficult and costly knowledge acquisition, may fail on cases outside its rules.

Architecture of an Expert System

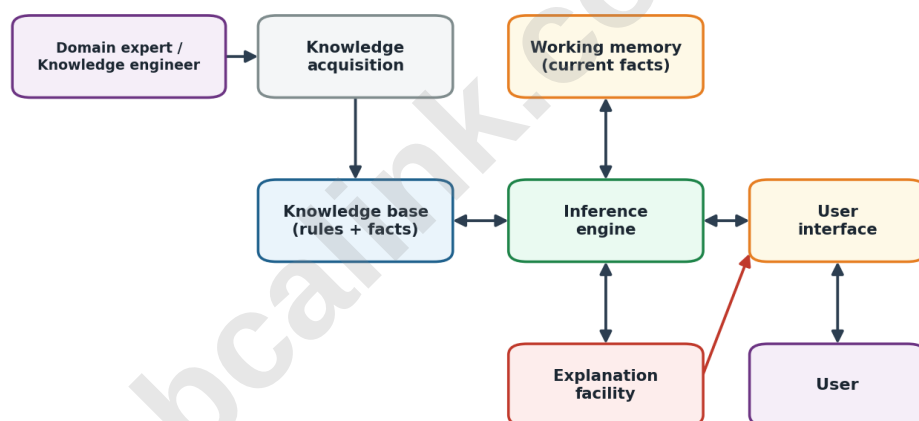


Figure 6.1 — Architecture of an expert system

Component	Function
Knowledge base	Stores domain knowledge as facts and rules (long-term memory); the heart of the system
Inference engine	Reasoning mechanism: matches facts against rules and derives conclusions by forward chaining (data-driven) or backward chaining (goal-driven); resolves conflicts when several rules apply
Working memory	Holds the facts of the current case (user answers, intermediate results)
User interface	Lets the user enter facts/questions and view advice, usually in a dialogue
Explanation facility	Answers “Why do you ask this?” and “How did you reach this conclusion?” by showing the rules used

Component	Function
Knowledge acquisition module	Helps the knowledge engineer enter and update expert knowledge

People involved: the **domain expert** (source of knowledge), the **knowledge engineer** (collects and encodes the knowledge) and the **end user**. An **expert system shell** (e.g., CLIPS, Jess; EMYCIN from MYCIN) is an ES without domain knowledge: only the knowledge base must be added.

Example — A small diagnostic ES (car will not start)

R1: IF the engine does not crank AND the lights are dim THEN the battery is weak.

R2: IF the battery is weak THEN recharge or replace the battery.

R3: IF the engine cranks AND there is no fuel smell THEN check the fuel supply.

Forward chaining: facts “engine does not crank”, “lights dim” → R1 gives “battery weak” → R2 gives the advice. **Backward chaining:** goal “recharge battery?” → needs “battery weak” → needs “no crank” and “dim lights”, asked from the user.

Handling Uncertainty — Certainty Factors (MYCIN style)

Each rule and fact carries a **certainty factor (CF)** from -1 (certainly false) to $+1$ (certainly true).

- $CF(A \text{ AND } B) = \min(CF(A), CF(B))$; $CF(A \text{ OR } B) = \max(CF(A), CF(B))$.
- $CF \text{ of a conclusion} = CF(\text{premise}) \times CF(\text{rule})$.
- Two rules supporting the same hypothesis (both positive): $CF = CF_1 + CF_2 \times (1 - CF_1)$.

Example

Rule 1 gives hypothesis H with CF 0.6, and Rule 2 gives H with CF 0.5. Combined $CF = 0.6 + 0.5 \times (1 - 0.6) = 0.8$.

Stages (Phases) of Expert System Development

Stage	Activities
1. Problem identification and feasibility	Decide whether an ES is suitable (narrow, knowledge-intensive task; an expert exists; benefits justify cost); define scope and goals
2. Knowledge acquisition	The knowledge engineer collects knowledge through interviews, observation, case studies, manuals and textbooks

Stage	Activities
3. Knowledge representation (design)	Choose how to represent knowledge (rules, frames, semantic nets) and the inference strategy; select a shell or programming language
4. Implementation (prototype)	Encode the knowledge base, build the inference engine and user interface; develop a working prototype
5. Testing, verification and validation	Run test cases; compare results with the human expert; find and correct errors and gaps in the knowledge
6. Deployment and integration	Install the system, train users, integrate it with existing systems and data
7. Maintenance and evolution	Keep the knowledge up to date, fix errors, and extend the system as the domain changes

*Note: Expert systems are built by **incremental prototyping**: a small prototype is developed, tested with the expert, refined, and extended, repeating stages 2–5 many times.*

6.2 Machine Vision

Concept of Machine Vision

Machine vision gives machines the ability to **acquire, process, analyze and understand images** and use the result to make decisions or control actions, much as human sight and the visual cortex do. It is the engineering application of computer vision, widely used in industry for inspection and robot guidance.

Field	Input → Output	Purpose
Image processing	Image → improved image	Enhance or transform images (noise removal, sharpening)
Computer vision	Image → description / understanding	Recognize objects, scenes and activity
Machine vision	Image → decision / action	Automated inspection, measurement, robot control in real time

Steps of Machine Vision

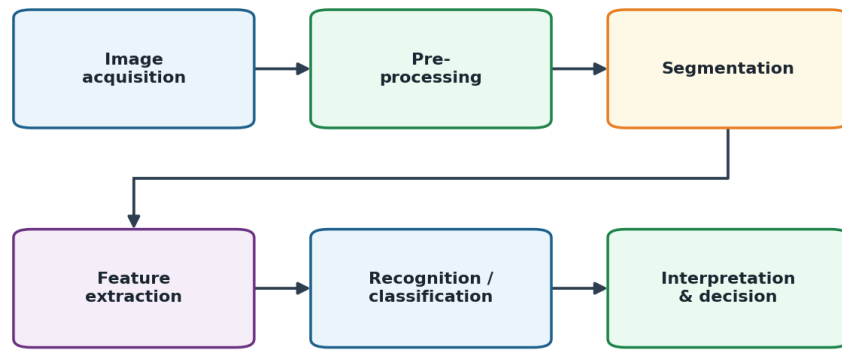


Figure 6.2 — Steps of a machine vision system

- **1. Image acquisition:** a camera or sensor captures the scene under controlled lighting and converts it into a digital image — a grid of **pixels**, obtained by *sampling* (spatial resolution) and *quantization* (gray levels).
- **2. Pre-processing:** improves image quality — noise removal (filtering), contrast enhancement, gray-scale conversion, resizing, normalization.
- **3. Segmentation:** separates objects from the background by **thresholding**, **edge detection** (Sobel, Canny), region growing or clustering.
- **4. Feature extraction:** measures useful properties of segmented regions — edges, corners, shape, size, texture, colour.
- **5. Recognition / classification:** identifies objects by matching features with templates or by a classifier (e.g., a neural network).
- **6. Interpretation and decision:** understands what has been recognized and triggers an action: accept/reject a part, steer a robot, raise an alarm.

Example — Thresholding and edge detection

Thresholding: choose $T = 128$. A pixel with gray level ≥ 128 becomes 1 (object), otherwise 0 (background). Pixel values 40, 200, 130, 90 $\rightarrow$ 0, 1, 1, 0.

Sobel edge detection: convolve the image with $G_x = [-1 \ 0 \ 1; -2 \ 0 \ 2; -1 \ 0 \ 1]$ and $G_y = [-1 \ -2 \ -1; 0 \ 0 \ 0; 1 \ 2 \ 1]$. The gradient magnitude $\sqrt{G_x^2 + G_y^2}$ is large at edges, where brightness changes rapidly.

Example — Bottle-cap inspection on a production line

Camera captures the cap $\rightarrow$ noise filtering $\rightarrow$ threshold separates the cap from the conveyor $\rightarrow$ measure diameter and detect the logo $\rightarrow$ classifier decides whether the cap is **good or defective** $\rightarrow$ a mechanical arm rejects defective caps.

Applications of Machine Vision

Area	Applications
Industry	Quality inspection, defect detection, dimension measurement, assembly guidance, barcode reading
Medicine	Analysis of X-ray, CT and MRI images; cell counting; surgical assistance
Transportation	Self-driving vehicles, lane and traffic-sign detection, number-plate recognition
Security / biometrics	Face, fingerprint and iris recognition; surveillance
Agriculture	Fruit grading, weed and disease detection, crop monitoring by drones
Robotics	Object picking, navigation, obstacle avoidance
Document processing	OCR — reading printed and handwritten text
Remote sensing	Satellite image analysis for land use, weather and disaster assessment

Challenges: varying illumination, occlusion, different viewpoints and scale, noise, recovering 3-D information from 2-D images, and meeting real-time speed requirements.

Key Takeaways

- ✓ An expert system emulates a human expert using a knowledge base, inference engine, working memory, explanation facility, user interface and knowledge acquisition module.
- ✓ Roles: domain expert, knowledge engineer, user. Uncertainty can be handled with certainty factors.
- ✓ Development stages: problem identification → knowledge acquisition → representation/design → implementation → testing → deployment → maintenance.
- ✓ Machine vision converts images into decisions: acquisition → pre-processing → segmentation → feature extraction → recognition → interpretation/decision.
- ✓ Applications: inspection, medical imaging, autonomous vehicles, biometrics, OCR, agriculture and robotics.

Review Questions

- What is an expert system? Draw and explain its architecture.
- Explain the stages of expert system development.
- How does an expert system differ from a conventional program? Give two examples of expert systems.
- Define machine vision. Explain its steps with a diagram.
- List applications of machine vision and its main challenges.

— *End of Unit 6* —

Reminder: these notes may contain errors — verify with your textbook and teacher.

bcalink.com