UNIT-I
Content

Introduction: Introduction to Artificial Intelligence, Foundations and History of Artificial Intelligence, Applications of Artificial Intelligence, Intelligent Agents, Structure of Intelligent Agents. Computer vision, Natural Language Possessing.

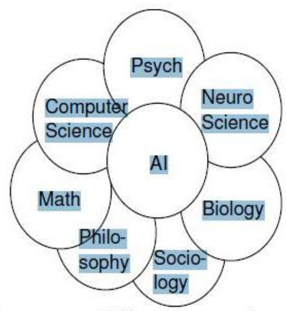
Introduction to Artificial Intelligence:

What is Artificial Intelligence?

There are many definitions;

- **Systems that think like humans**
1. Machines with minds, in the full and literal sense
- **Systems that think rationally**
1. The study of mental faculties through the use of computational models.
2. The study of the computations that make it possible to perceive, reason, and act.
- **Systems that act like humans**
1. The study of how to make computers do things that, at the moment, people are better.
2. The art of creating machines that performs functions that require intelligence when performed by people.
- **Systems that act rationally**
1. Computational intelligence is the study and design of intelligent agents.
2. Intelligent behavior in artifacts

AI has many intersections with other disciplines, and many approaches to the AI problem



We will draw from many different areas that contribute to AI.

1. Systems that think like humans

Most closely related to the field of cognitive science. We need to get inside the actual workings of the human mind and implement this in the computer. One approach is by psychological experiment, the other by introspection. Still another is biologically to reconstruct a computer brain in the same manner as human brains.

2. Systems that act humanely

Under this approach the goal is to create a system that acts the same way that humans do, butmay be implemented in a totally different way. We'll see the Turing Test shortly which is a way to determine if a system achieves the goal of acting humanely without regard to internal representations.

For example, a system might appear to act like a human by inserting random typing errors, but doesn't actually make errors the same way that a human would.

3. Systems that think rationally

There is a tradition of using the "laws of thought" that dates back to Socrates and Aristotle. Their study initiated the field of logic. The logic is the tradition within AI hopes to build on this approach to create intelligent systems; the main problem has been scaling this approach up beyond toy systems.

4. Systems that act rationally

An agent is something that acts. To distinguish an agent from any other program it is intended to perceive its environment, adapt to change, and operate autonomously. A rational agent is one that acts to achieve the best outcome, or best expected outcome when there is uncertainty. Unlike the "laws of thought" approach, these agents might act on incomplete knowledge or to still act when it is not possible to prove what the correct thing to do is. This approach makes it more general than the "laws of thought" approach and more amenable to scientific development than the pure "human-based" approach.

"A **rational agents** is one that acts so as to achieve the best outcome or, when there is uncertainty, the best expected outecome**"**

Agent-based activity has focused on the issues of:

1) Autonomy. Agents should be independent and communicate with others as necessary.
2) Situated. Agents should be sensitive to their own surroundings and context.
3) Interactional. Often an interface with not only humans, but also with other agents.
4) Structured. Agents cooperate in a structured society.
5) Emergent. Collection of agents more powerful than an individual agent.

Another way to think about the field of AI is in term of task domains:

1. **Mundane:**
   Vision, Speech
   Natural Language Processing, Generation, Understanding
   Reasoning
   Motion
2. **Formal:**
   Board Game-Playing, chess, checkers, gobblet
   Logic
   Calculus
   Algebra
   Verification, Theorem Proving
3. **Expert:**
   Design, engineering, graphics
   Art, creativity
   Music
   Financial Analysis
   Consulting

- **Classification of AI is Weak vs. Strong AI:**
  This is essentially the human vs. non-human approach.

1) Weak AI. The study and design of machines that perform intelligent tasks. Not concerned with how tasks are performed, mostly concerned with performance and efficiency, such as solutions that are reasonable for NP-Complete problems. E.g., to make a flying machine, use logic and physics, don't mimic a bird.

2) Strong AI. The study and design of machines that simulate the human mind to perform intelligent tasks. Borrow many ideas from psychology, neuroscience. Goal is to perform tasks the way a human might do them – which makes sense, since we do have models of human thought and problem solving. Includes psychological ideas in STM, LTM, forgetting, language, genetics, etc. Assumes that the physical symbol hypothesis holds.

3) Evolutionary AI. The study and design of machines that simulate simple creatures, and attempt to evolve and have higher level emergent behavior. For example, ants, bees, etc.

Theoretical Foundations of AI

- **Philosophy -** Logic methods of reasoning, mind as physical system foundations of learning, language, rationality
- **Mathematics -** Formal representation and proof algorithms, computation, (un)decidability, (in)tractability, probability
- **Economics -** Utility, decision theory, game theory.
- **Neuroscience -** Physical substrate for mental activity
- **Psychology -** Phenomena of perception and motor control, experimental techniques
- **Computer Engineering -** Building fast computers
- **Control theory-** Design systems that maximize an objective function over time
- **Linguistics -** Knowledge representation, grammar

Abridged History of AI

1943 McCulloch & Pitts: Boolean circuit model of brain
1950 Turing's "Computing Machinery and Intelligence"
1956 Dartmouth meeting: "Artificial Intelligence" adopted
1952—69 Look, Ma, no hands!
1950s Early AI programs, including Samuel's checkers program, Newell & Simon's Logic Theorist, Gelernter's Geometry Engine
1965 Robinson's complete algorithm for logical reasoning
1966—73 AI discovers computational complexity Neural network research almost disappears
1969—79 Early development of knowledge-based systems
1980-- AI becomes an industry
1986-- Neural networks return to popularity
1987-- AI becomes a science
1995-- The emergence of intelligent agents, genetic algorithms

AI Applications

Few applications of artificial intelligence.
1. Game-playing. IBM's deep-blue has beaten Kasparov, and we have a world-champion caliber Backgammon program.

2. Automated reasoning and theorem-proving.

3. Expert Systems. An expert system is a computer program with deep knowledge in a specific niche area that provides assistance to a user. Famous examples include DENDREAL, an expert system that inferred the structure of organic molecules from their spectrographic information, and MYCIN.

4. Machine Learning. Systems that can automatically classify data and learn from new examples has become more popular, especially as the Internet has grown and spawned applications that require personalized agents to learn a user's interests.

5. Natural Language Understanding, Semantic Modeling.

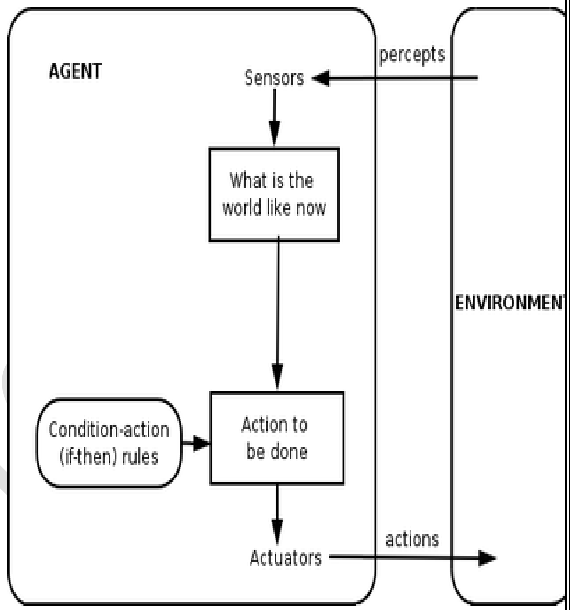
6. Modeling Human Performance. As described earlier, machine intelligence need not pattern itself after human intelligence.

7. Planning and Robotics. Planning research began as an effort to design robots that could perform their task. For example, the Sojourner robot on Mars was able to perform some of its own navigation tasks since the time delay to earth makes real-time control impossible.

8. Languages and Environments. LISP and PROLOG were designed to help support AI, along with constructs such as object-oriented design and knowledge bases.

Intelligent agent

- In artificial intelligence, an **intelligent agent** (**IA**) is an autonomous entity which observes through sensors and acts upon an environment using actuators (i.e. it is an agent) and directs its activity towards achieving goals (i.e. it is rational). Intelligent agents may also learn or use knowledge to achieve their goals.
- They may be very simple or very complex: a reflex machine such as a thermostat is an intelligent agent, as is a human being, as is a community of human beings working together towards a goal.
- Intelligent agents are often described schematically as an abstract functional system similar to a computer program. For this reason, intelligent agents are sometimes called **abstract intelligent agent**s (AIA) to distinguish them from their real world implementations as computer systems, biological systems, or organizations. Some definitions of intelligent agents emphasize their autonomy, and so prefer the term **autonomous intelligent agents**. Still others (notably Russell & Norvig (2003)) considered goal-directed behavior as the essence of intelligence and so prefer a term borrowed from economics, "rational agent".
- Intelligent agents in artificial intelligence are closely related to agents in economics, and versions of the intelligent agent paradigm are studied in cognitive science, ethics, the philosophy of practical reason, as well as in many interdisciplinary socio-cognitive modeling and computer social simulations.



Simple Reflex Agent

- Intelligent agents are also closely related to software agents (an autonomous computer program that carries out tasks on behalf of users).

The Structure of Agents

- A simple agent program can be defined mathematically as an agent function which maps every possible percepts sequence to a possible action the agent can perform or to a coefficient, feedback element, function or constant that affects eventual actions:

$$f : P^* \rightarrow A$$

- Agent function is an abstract concept as it could incorporate various principles of decision making like calculation of utility of individual options, deduction over logic rules, fuzzy logic, etc. The **program agent**, instead, maps every possible percept to an action.
- The term percept to refer to the agent's perceptional inputs at any given instant. In the following figures an agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.

Classes of intelligent agents

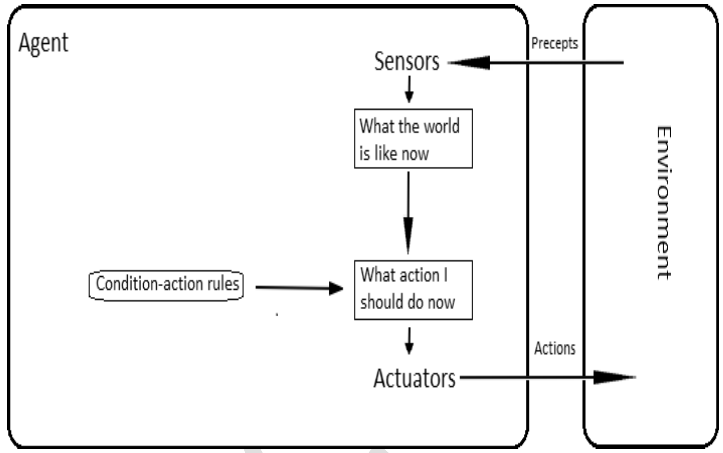
Russell & Norvig group agents into five classes based on their degree of perceived intelligence and capability

1. simple reflex agents
2. model-based reflex agents
3. goal-based agents

4. utility-based agents
5. learning agents

Simple reflex agents

- Simple reflex agents act only on the basis of the current percept, ignoring the rest of the percept history. The agent function is based on the *condition-action rule*: if condition then action.
- This agent function only succeeds when the environment is fully observable. Some reflex agents can also contain information on their current state which allows them to disregard conditions whose actuators are already triggered.
- Infinite loops are often unavoidable for simple reflex agents operating in partially observable environments. Note: If the agent can randomize its actions, it may be possible to escape from infinite loops.



function SIMPLE-REFLEX-AGENT(percept) returns action

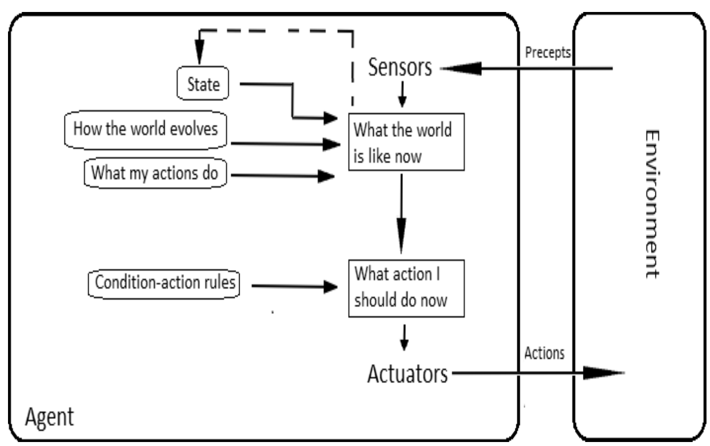
static: rules, a set of condition-action rules
state ← INTERPRET-INPUT(percept)
rule ← RULE-MATCH(state,rules)
action ← RULE-ACTION[rule]
return action

" A simple reflex agent. It works by finding a rule whose condition matches the current situation (as defined by the percept) and then doing the action associated with that rule."

Model-based reflex agents

- A model-based agent can handle a partially observable environment. Its current state is stored inside the agent maintaining some kind of structure which describes the part of the world which cannot be seen. This knowledge about "how the world works" is called a model of the world, hence the name "model-based agent".

- A model-based reflex agent should maintain some sort of internal model that depends on the percept history and thereby reflects at least some of the unobserved aspects of the current state. It then chooses an action in the same way as the reflex agent.



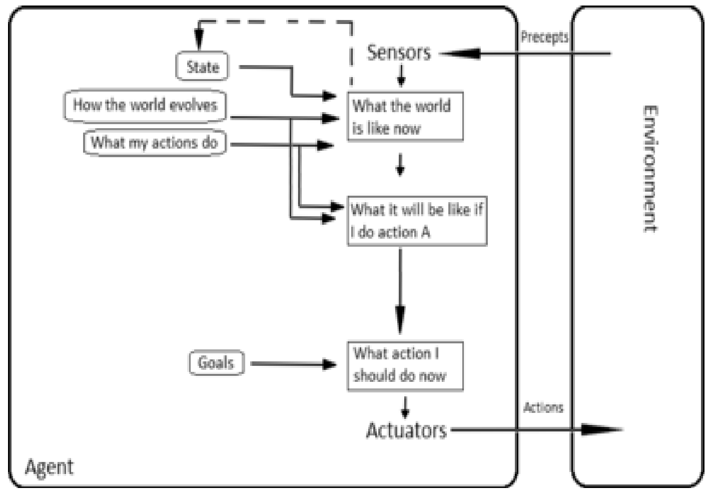
function REFLEX-AGENT-WITH-STATE(*percept*) **returns** *action*
static: *state*, a description of the current world state
rules, a set of condition-action rules
action, the most recent action initially none

state ← UPDATE-STATE(*state, percept*)
rule ← RULE-MATCH(*state, rules*)

action ←RULE-ACTION[*rule*]
state ← UPDATE-STATE(*state*, *action*)
return *action*

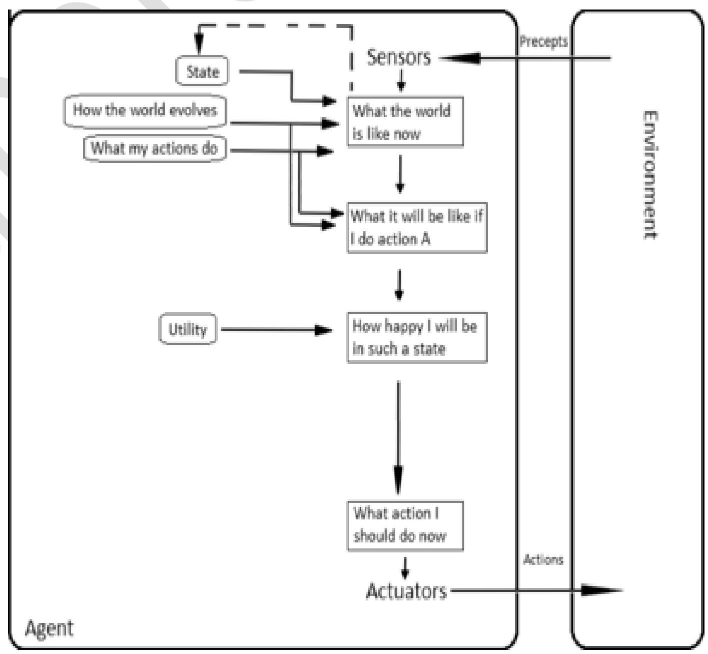
Goal-based agents

- Goal-based agents further expand on the capabilities of the model-based agents, by using "goal" information.
- Goal information describes situations that are desirable. This allows the agent a way to choose among multiple possibilities, selecting the one which reaches a goal state.
- Search and planning are the subfields of artificial intelligence devoted to finding action sequences that achieve the agent's goals.
- In some instances the goal-based agent appears to be less efficient; it is more flexible because the knowledge that supports its decisions is represented explicitly and can be modified.



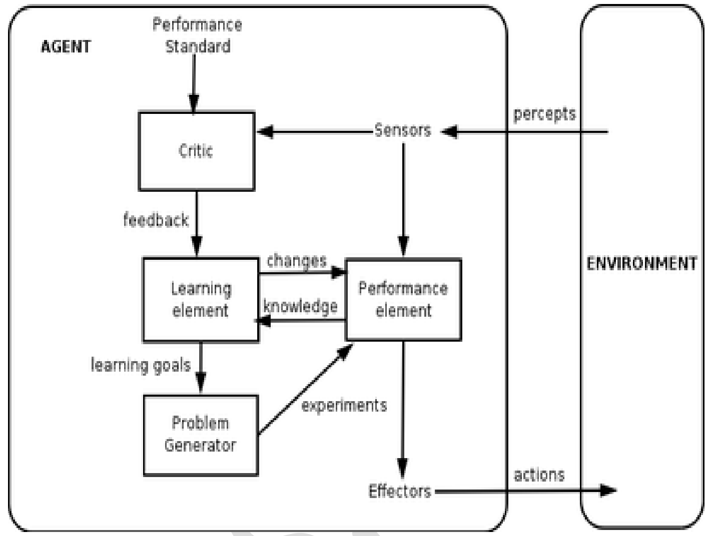
Utility-based agents

- Goal-based agents only distinguish between goal states and non-goal states.
- It is possible to define a measure of how desirable a particular state is. This measure can be obtained through the use of a *utility function* which maps a state to a measure of the utility of the state.
- A more general performance measure should allow a comparison of different world states according to exactly how happy they would make the agent.
- The term utility can be used to describe how "happy" the agent is.
- A utility-based agent has to model and keep track of its environment, tasks that have involved a great deal of research on perception, representation, reasoning, and learning.



Learning agents

- Learning has an advantage that it allows the agents to initially operate in unknown environments and to become more competent than its initial knowledge alone might allow.
- The most important distinction is between the "learning element", which is responsible for making improvements, and the "performance element", which is responsible for selecting external actions.
- The learning element uses feedback from the "critic" on how the agent is doing and determines how the performance element should be modified to do better in the future.
- The performance element is what we have previously considered to be the entire agent: it takes in percepts and decides on actions.
- The last component of the learning agent is the "problem generator". It is responsible for suggesting actions that will lead to new and informative experiences.



Other classes of intelligent agents

According to other sources ,some of the sub-agents that may be a part of an Intelligent Agent or a complete Intelligent Agent in themselves are:

- Decision Agents
- Input Agents (that process and make sense of sensor inputs – e.g. neural network based agents);
- Processing Agents (that solve a problem like speech recognition);
- Spatial Agents (that relate to the physical real-world);
- World Agents (that incorporate a combination of all the other classes of agents to allow autonomous behaviors).
- Believable agents - An agent exhibiting a personality via the use of an artificial character (the agent is embedded) for the interaction.
- Physical Agents - A physical agent is an entity which *percepts* through sensors and *acts* through actuators.
- Temporal Agents - A temporal agent may use time based stored information to offer instructions or data *acts* to a computer program or human being and takes program inputs *percepts* to adjust its next behaviors.

Natural language processing

- **Natural language processing** (**NLP**) is a field of computer science, artificial intelligence, and linguistics concerned with the interactions between computers and human (natural) languages.
- Many challenges in NLP involve natural language understanding, that is, enabling computers to derive meaning from human or natural language input, and others involve natural language generation.

NLP using machine learning

- Modern NLP algorithms are based on machine learning, especially statistical machine learning.
- The machine-learning paradigm calls instead for using general learning algorithms — often, although not always, grounded in statistical inference — to automatically learn such rules through the analysis of large *corpora* of typical real-world examples.
- Many different classes of machine learning algorithms have been applied to NLP tasks.
- Some of the earliest-used algorithms, such as decision trees, produced systems of hard if-then rules similar to the systems of hand-written rules that were then common.

Major tasks in NLP

A list of some of the most commonly researched tasks in NLP

Automatic summarization

Produce a readable summary of a chunk of text. Often used to provide summaries of text of a known type, such as articles in the financial section of a newspaper.

Machine translation

Automatically translate text from one human language to another.

Morphological segmentation

Separate words into individual morphemes and identify the class of the morphemes.

Named entity recognition (NER)

Given a stream of text, determine which items in the text map to proper names, such as people or places, and what the type of each such name is

Natural language generation

Convert information from computer databases into readable human language.

Natural language understanding

Convert chunks of text into more formal representations such as first-order logic structures that are easier for computer programs to manipulate. Natural language understanding involves the identification of the intended semantic from the multiple possible semantics which can be derived from a natural language expression which usually takes the form of organized notations of natural languages concepts.

Optical character recognition (OCR)

Given an image representing printed text, determine the corresponding text.

Part-of-speech tagging

Given a sentence, determine the part of speech for each word. Many words, especially common ones, can serve as multiple parts of speech.

Parsing

Determine the parse tree (grammatical analysis) of a given sentence.The grammar for natural lnguages is ambiguous and typical sentences have multiple possible analyses.

Question answering

Given a human-language question, determine its answer.

Relationship extraction

Given a chunk of text, identify the relationships among named entities.

Speech recognition

Given a sound clip of a person or people speaking, determine the textual representation of the speech. This is the opposite of text to speech and is one of the extremely difficult problems colloquially termed "AI-complete"

Speech segmentation

Given a sound clip of a person or people speaking, separate it into words.

Information retrieval (IR)

This is concerned with storing, searching and retrieving information. It is a separate field within computer science (closer to databases), but IR relies on some NLP methods (for example, stemming).

Information extraction (IE)

This is concerned in general with the extraction of semantic information from text. This covers tasks such as named entity recognition, Coreference resolution, relationship extraction, etc.

Computer vision

- **Computer vision** is a field that includes methods for acquiring, processing, analyzing, and understanding images and, in general, high-dimensional data from the real world in order to produce numerical or symbolic information, *e.g.*, in the forms of decisions.
- A theme in the development of this field has been to duplicate the abilities of human vision by electronically perceiving and understanding an image.
- Computer vision has also been described as the enterprise of automating and integrating a wide range of processes and representations for vision perception.

As a technological discipline, computer vision seeks to apply its theories and models to the construction of computer vision systems. Examples of applications of computer vision include systems for:

- Controlling processes, *e.g.*, an industrial robot;
- Navigation, *e.g.*, by an autonomous vehicle or mobile robot;
- Detecting events, *e.g.*, for visual surveillance or people counting;
- Organizing information, *e.g.*, for indexing databases of images and image sequences;
- Modeling objects or environments, *e.g.*, medical image analysis or topographical modeling;
- Interaction, *e.g.*, as the input to a device for computer-human interaction, and
- Automatic inspection, *e.g.*, in manufacturing applications.

Computer vision system methods

The organization of a computer vision system is highly application dependent. The specific implementation of a computer vision system also depends on if its functionality is pre-specified or if some part of it can be learned or modified during operation. Many functions are unique to the application. Typical functions which are found in many computer vision systems.

- **Image acquisition** – A digital image is produced by one or several image sensors, which, besides various types of light-sensitive cameras, include range sensors, tomography devices, radar, ultra-sonic cameras, etc. Depending on the type of sensor, the resulting image data is an ordinary 2D image, a 3D volume, or an image sequence.
- **Pre-processing** – Before a computer vision method can be applied to image data in order to extract some specific piece of information, it is usually necessary to process the data in order to assure that it satisfies certain assumptions implied by the method. Examples are
  - Re-sampling in order to assure that the image coordinate system is correct.
  - Noise reduction in order to assure that sensor noise does not introduce false information.
  - Contrast enhancement to assure that relevant information can be detected.
  - **Feature extraction** – Image features at various levels of complexity are extracted from the image data.
- **Detection/segmentation** – At some point in the processing a decision is made about which image points or regions of the image are relevant for further processing.
  - Selection of a specific set of interest points
  - Segmentation of one or multiple image regions which contain a specific object of interest.

- **High-level processing** – At this step the input is typically a small set of data, for example a set of points or an image region which is assumed to contain a specific object. The remaining processing deals with, for example:
    - Verification that the data satisfy model-based and application specific assumptions.
    - Estimation of application specific parameters, such as object pose or object size.
    - Image recognition – classifying a detected object into different categories.
    - Image registration – comparing and combining two different views of the same object.
- **Decision making** Making the final decision required for the application, for example:
    - Pass/fail on automatic inspection applications
    - Match / no-match in recognition applications
    - Flag for further human review in medical, military, security and recognition applications