

Introduction to Computer Vision and Image Processing Fundamentals

Ogban Ugot



Content

1. Defining Artificial Intelligence
2. Why Computer Vision?
3. Image Processing with Python and OpenCV

What is AI?

Defining AI — A Multi-Dimensional Approach

AI can be defined along two primary dimensions: **thinking vs. acting** and **human-like vs. ideal (rational)** behavior. These give rise to four categories:

- **Thinking Humanly:** Mimicking human cognitive processes.
- **Thinking Rationally:** Emulating correct, logical reasoning.
- **Acting Humanly:** Imitating human behavior in observable actions.
- **Acting Rationally:** Making optimal decisions based on knowledge and expected outcomes.

Each perspective offers unique insights and has shaped different subfields and methodologies within AI research.

Acting Humanly — The Turing Test Approach

Proposed by Alan Turing in 1950, the Turing Test evaluates whether a computer can mimic human responses so convincingly that a human interrogator cannot distinguish it from a person. To pass the Turing Test, a computer must demonstrate:

- **Natural Language Processing:** To understand and respond in human language.
- **Knowledge Representation:** To store and retrieve facts and beliefs.
- **Automated Reasoning:** To draw logical conclusions and answer questions.
- **Machine Learning:** To adapt and improve with experience.

Although the Turing Test has symbolic significance, few researchers focus solely on passing it today.

Beyond the Turing Test — The Total Turing Test

The Total Turing Test extends the original concept by including perceptual and physical tasks:

- **Computer Vision:** To visually interpret the world.
- **Robotics:** To physically interact with and navigate environments.
- Turing's test remains influential not because it defines AI's endpoint, but because it highlights essential competencies required for general intelligence. Modern researchers, however, prioritize understanding core principles over mimicking humans perfectly.

Thinking Humanly — The Cognitive Modeling Approach

To say a program thinks like a human requires understanding human cognition itself. Cognitive modeling uses:

- **Introspection:** Examining one's own thought processes.
- **Psychological Experiments:** Observing human behavior in controlled settings.
- **Neuroimaging:** Studying brain activity.

By developing computational models that simulate human thought patterns, researchers can test theories about cognition. The interdisciplinary field of **cognitive science** merges AI and psychology to create testable, mechanistic explanations of mental processes.

Thinking Rationally — The “Laws of Thought” Approach

Dating back to Aristotle, this approach centers on codifying perfect reasoning:

- **Syllogisms**: Structured arguments that yield true conclusions from true premises.
- **Logic**: A formal language for expressing and manipulating knowledge.
- 19th-century logicians laid the groundwork for programs that could solve problems expressed in logical form. However, translating human knowledge into formal logic is challenging, and reasoning systems often struggle with computational complexity. These issues gave rise to more flexible AI paradigms that incorporate uncertainty and practical constraints.

Acting Rationally — The Rational Agent Approach

The rational agent model defines an AI as an **agent** that perceives its environment and acts to maximize expected outcomes. It encompasses:

- **Autonomy**: Operating independently.
- **Adaptability**: Learning and adjusting to new circumstances.
- **Goal-orientation**: Pursuing objectives intelligently.
- Rationality includes logical inference but also reflexes and heuristics that bypass explicit reasoning when necessary. This framework is more general and scientifically tractable than human-centered approaches, making it the dominant paradigm in AI today.

Rationality and Real-World Constraints

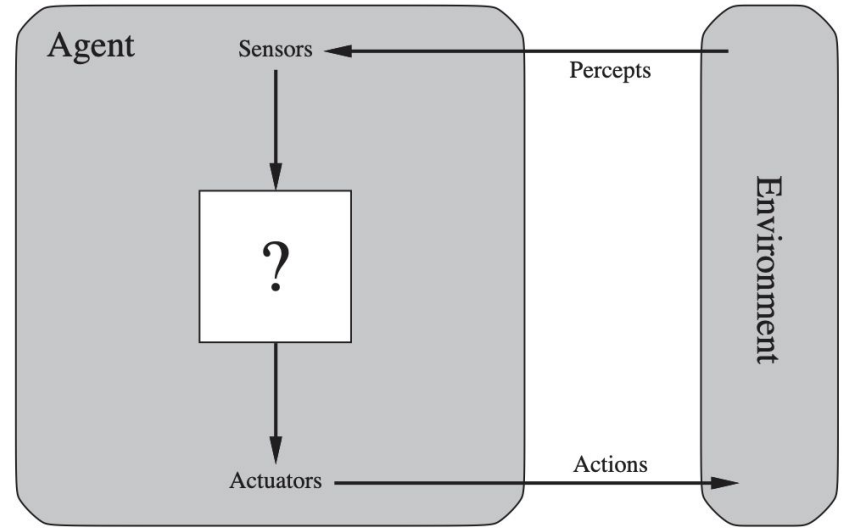
- While rationality is a compelling ideal, real-world environments are too complex for perfect rationality.
- Computational limitations and incomplete information force systems to adopt bounded rationality—acting effectively under constraints.
- Modern AI uses the **rational agent** as a foundation while acknowledging that trade-offs and approximations are inevitable.
- The journey toward **general intelligence** involves embracing this complexity while designing agents that make good decisions in practice.

Building a Rational Agent

What Makes an Agent?

Core Components and Characteristics

- **Agent:** Anything that perceives its environment through sensors and acts through actuators
- **Environment:** The world in which the agent operates
- **Sensors:** Mechanisms for perceiving the environment
- **Actuators:** Mechanisms for acting upon the environment
- **Coupling:** The interaction between agent and environment determines behavior



Agent Examples Across Domains

From Humans to Software

Human Agent:

- Sensors: Eyes, ears, other organs
- Actuators: Hands, legs, vocal tract

Robotic Agent:

- Sensors: Cameras, infrared range finders
- Actuators: Various motors

Software Agent:

- Sensors: Keystrokes, file contents, network packets
- Actuators: Screen display, file writing, network transmission

Percepts and Percept Sequences

How Agents Process Information

- **Percept:** Agent's perceptual input at any given instant
- **Percept Sequence:** Complete history of everything the agent has ever perceived
- Agent's action choice can depend on entire percept sequence observed to date
- Actions cannot depend on unperceived information
- This forms the foundation for agent behavior analysis

Why Computer Vision?

The Perception Problem

Rational Agents Need Rich Environmental Input

Remember our rational agent framework:

- **Percepts:** Agent's perceptual input at any given instant
- **Sensors:** Mechanisms for perceiving the environment
- **Actions:** Cannot depend on unperceived information

The Challenge: Most real-world environments are primarily visual. Without computer vision, our agents are essentially blind to the richest source of environmental information.

Essential for General Intelligence

The Total Turing Test Connection

Original Turing Test Requirements:

- Natural Language Processing
- Knowledge Representation
- Automated Reasoning
- Machine Learning

Total Turing Test Additions:

- **Computer Vision** ← Critical!
- Robotics

Computer vision was recognized as essential for true AI because it enables machines to "**visually interpret the world**" - a fundamental requirement for general intelligence.

Expanding Agent Capabilities

From Limited to Rich Environmental Understanding

Without Computer Vision:

- Agents limited to text, audio, or simple sensor inputs
- Cannot understand spatial relationships
- Miss critical environmental context
- Restricted to narrow, specialized domains

With Computer Vision:

- Access to rich visual information (colors, shapes, movement, depth)
- Understanding of spatial relationships and object interactions
- Real-time environmental awareness
- Foundation for autonomous navigation, manipulation, and decision-making

Where Computer Vision Enables Rational Agents



Autonomous Systems:

- Self-driving cars
- Delivery drones
- Manufacturing robots



Healthcare & Safety:

- Medical image analysis
- Security surveillance
- Quality control



Human-Computer Interaction:

- Gesture recognition
- Augmented reality
- Facial recognition

Where Computer Vision Enables Rational Agents...Contd.



Scientific Discovery:

- Astronomical imaging
- Microscopy analysis
- Environmental monitoring

Bottom Line: Computer vision transforms AI from narrow, text-based systems into agents that can perceive, understand, and act in our visual world.

Image Processing with Python & OpenCV

Background: OpenCV and NumPy

OpenCV (Open Source Computer Vision Library): https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html

- A powerful open-source library for **real-time computer vision**, **image processing**, and **machine learning**.
- Written in C/C++ but has **full Python bindings** (`cv2`).
- Supports:
 - Reading/writing images and videos
 - Image transformations (resizing, filtering, rotation)
 - Feature detection, object tracking, face recognition, etc.
 - Integration with deep learning models

Originally developed by Intel in 1999, now maintained by OpenCV.org.

NumPy (Numerical Python)

- A core Python library for **efficient numerical and matrix operations**. <https://numpy.org/doc/>
- Provides the `ndarray` object — a fast, flexible **multi-dimensional array** structure.
- Powers nearly all scientific computing in Python (used in pandas, scikit-learn, etc.).

How OpenCV and NumPy Are Related

- OpenCV uses **NumPy arrays** to represent images.
 - When you load an image using `cv2.imread()`, you get a **NumPy array** of shape `(H, W, 3)` for color images or `(H, W)` for grayscale.
- This means you can:
 - Slice, index, and modify images using standard NumPy syntax.
 - Apply mathematical operations (e.g., brightness scaling, masking, filtering).
- OpenCV functions are **optimized for speed**, but you can mix them freely with NumPy for custom operations.

Practical Session

Github Repo:

<https://github.com/Society-For-AI/AI-Skill-Accelerator-Computer-Vision>

Use the notebook in Google Colab:

<https://colab.research.google.com/drive/1OvO5BV6i7rqNhKemGyONJSeOvatNTvHc?usp=sharing>