AI history ….. problem solving using search

Main events in the history of AI

Period	**Key Events**
The birth of Artificial Intelligence (1943–1956)	McCulloch and Pitts, *A Logical Calculus of the Ideas Immanent in Nervous Activity*, 1943 Turing, *Computing Machinery and Intelligence*, 1950 The Electronic Numerical Integrator and Calculator project (von Neumann) Shannon, *Programming a Computer for Playing Chess*, 1950 The Dartmouth College summer workshop on machine intelligence, artificial neural nets and automata theory, 1956

Period	Key Events
The rise of artificial intelligence (1956–late 1960s)	*LISP* (McCarthy) The General Problem Solver (GPR) project (Newell and Simon) Newell and Simon, *Human Problem Solving*, 1972 Minsky, *A Framework for Representing Knowledge*, 1975
The disillusionment in artificial intelligence (late 1960s–early 1970s)	Cook, *The Complexity of Theorem Proving Procedures*, 1971 Karp, *Reducibility Among Combinatorial Problems*, 1972 The Lighthill Report, 1971

Period	**Key Events**
The discovery of expert systems (early 1970s–mid-1980s)	DENDRAL (Feigenbaum, Buchanan and Lederberg, Stanford University) MYCIN (Feigenbaum and Shortliffe, Stanford University) PROSPECTOR (Stanford Research Institute) PROLOG - *a logic programming language* (Colmerauer, Roussel and Kowalski, France) EMYCIN (Stanford University) Waterman, *A Guide to Expert Systems*, 1986

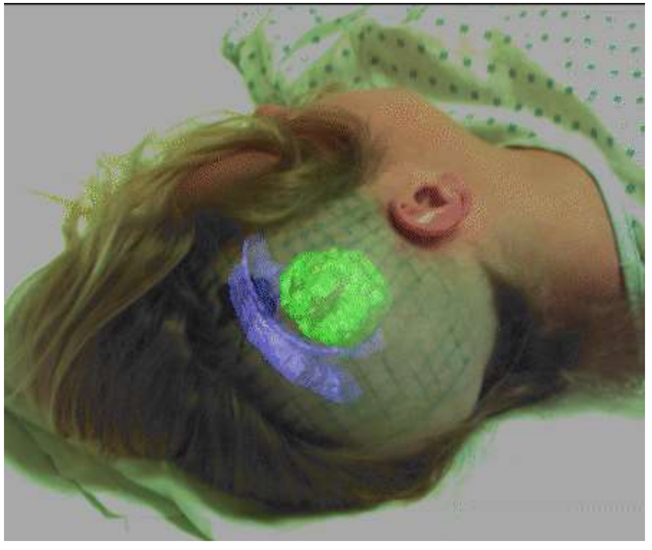
Period	**Key Events**
The rebirth of artificial neural networks (1965–onwards)	Hopfield, *Neural Networks and Physical Systems with Emergent Collective Computational Abilities*, 1982 Kohonen, *Self-Organized Formation of Topologically Correct Feature Maps*, 1982 Rumelhart and McClelland, *Parallel Distributed Processing*, 1986 The First IEEE International Conference on Neural Networks, 1987 Haykin, *Neural Networks*, 1994 Neural Network, MATLAB Application Toolbox (The MathWork, Inc.)

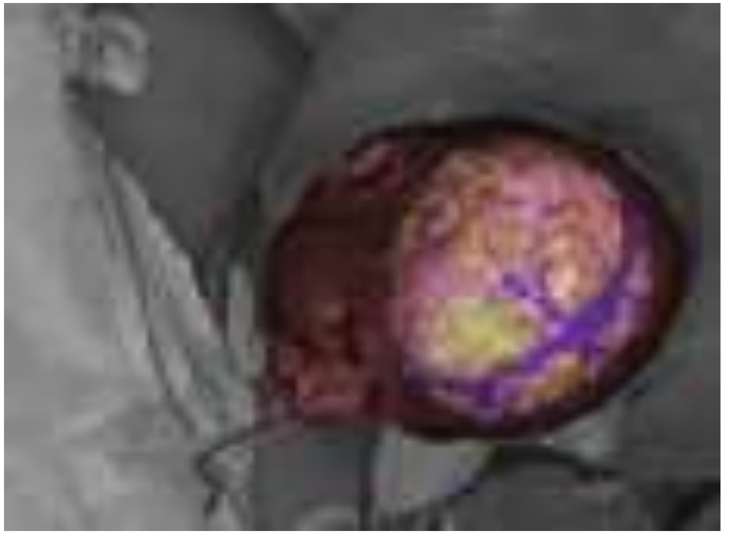
Period	**Key Events**
Evolutionary computation (early 1970s–onwards)	Rechenberg, *Evolutionsstrategien - Optimierung Technischer Systeme Nach Prinzipien der Biologischen Information*, 1973 Holland, *Adaptation in Natural and Artificial Systems*, 1975. Koza, *Genetic Programming: On the Programming of the Computers by Means of Natural Selection*, 1992. Schwefel, *Evolution and Optimum Seeking*, 1995 Fogel, *Evolutionary Computation –Towards a New Philosophy of Machine Intelligence*, 1995.

Period	**Key Events**
Computing with Words (late 1980s–onwards)	Zadeh, *Fuzzy Sets*, 1965 Zadeh, *Fuzzy Algorithms*, 1969 Mamdani, *Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis*, 1977 Sugeno, *Fuzzy Theory*, 1983 Japanese “fuzzy” consumer products (dishwashers, washing machines, air conditioners, television sets, copiers) Sendai Subway System (Hitachi, Japan), 1986 The First IEEE International Conference on Fuzzy Systems, 1992 Kosko, *Neural Networks and Fuzzy Systems*, 1992 Kosko, *Fuzzy Thinking*, 1993 Cox, *The Fuzzy Systems Handbook*, 1994 Zadeh, *Computing with Words - A Paradigm Shift*, 1996 Fuzzy Logic, MATLAB Application Toolbox (The MathWork, Inc.)

AI Applications

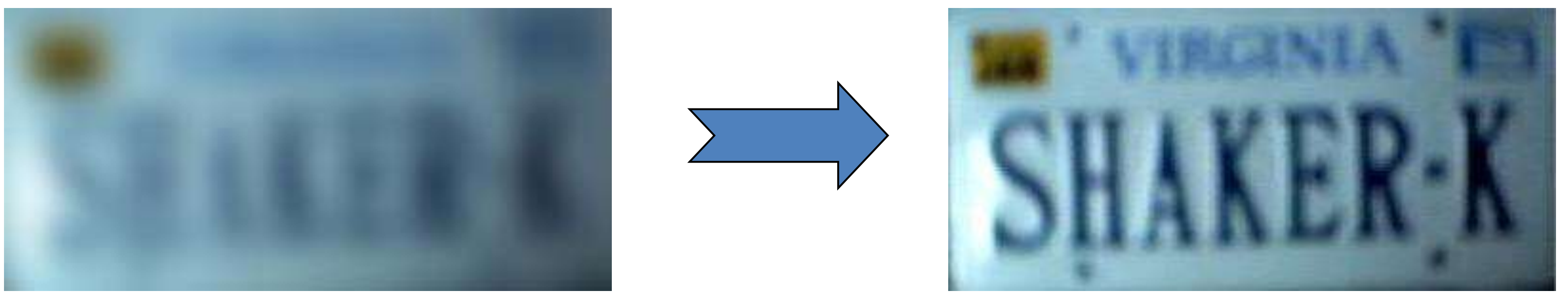
- **Medicine**:
  - Image guided surgery

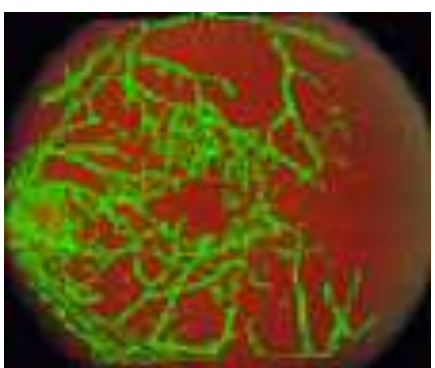


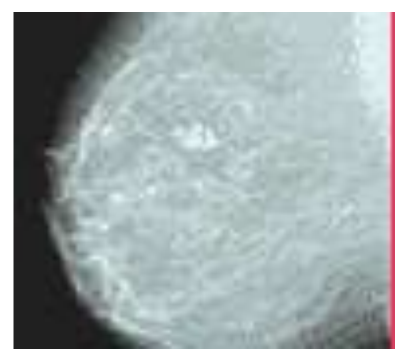


AI Applications

– Image analysis and enhancement







AI Applications

- **Transportation :Autonomous vehicle control:**

Pizza Delivery by Drone

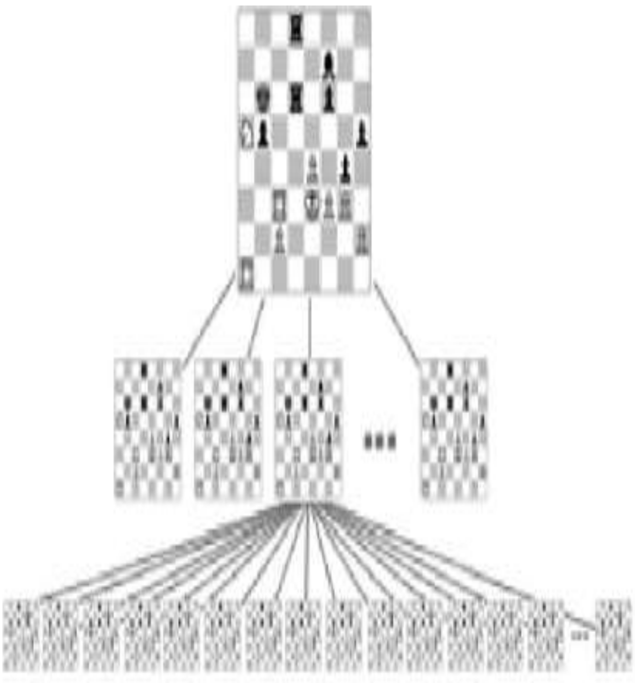
AI and the Future

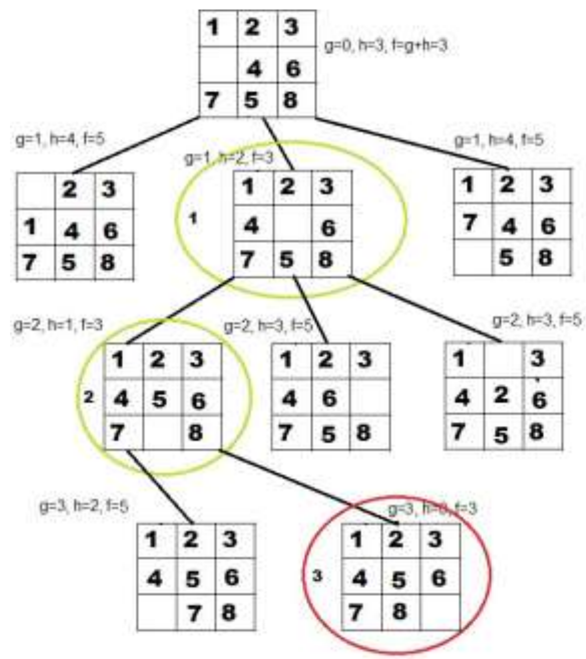
- Another Examples...

smart mirrors

speech recognition

1. Problem Solving using Search





What is search strategies ?

- A search strategy is defined by picking the order of node expansion – nodes are taken from the frontier • Strategies are evaluated along the following dimensions: –
- completeness: does it always find a solution if one exists?
- – time complexity: number of nodes generated
- – space complexity: maximum number of nodes in memory – optimality: does it always find a least-cost solution?
- • Time and space complexity are measured in terms of
- – b: maximum branching factor of the search tree
- – d: depth of the least-cost solution –
- m: maximum depth of the state space (may be ∞)

What kind of problem could be solved by search

- Many interesting problems in science and engineering are solved using search A search problem is defined by:
- • A search space: – The set of objects among which we search for the solution
- • A goal condition – Characteristics of the object we want to find in the search space?

- Search problems can be often represented as graph search problems:
- • Initial state – State (configuration) we start to search from (e.g. start city, initial game position)
- • Operators: – Transform one state to another (e.g. valid connections between cities, valid moves in Puzzle 8)
- • Goal condition: – Defines the target state (destination, winning position) Search space is now defined indirectly through: The initial state + Operators

BLIND Search Methods

Methods that do not use any specific knowledge about the problem:

Depth-first

Breadth-first

Non-deterministic search

Iterative deepening

Heuristic Search Methods

Methods that use a heuristic function to provide specific knowledge about the problem:

Hill climbing

Beam search

Hill climbing (2)

Greedy search

Genetic algorithm

OPTIMAL Search

When cost of TRAVERSING the path should be minimized (even at expense of more complicated SEARCHING) :

Uniform Cost

A*

Learning Outcomes

- To be able to apply all search algorithm in different problems like 8-puzzle or a simple roads network (hw#1)
- To know the differnces between the differnt type of algorithim (speed, memory
- To be able to compute the fitness or the Heuristic Functions for given problem
- To know what type of problems could be solve using search