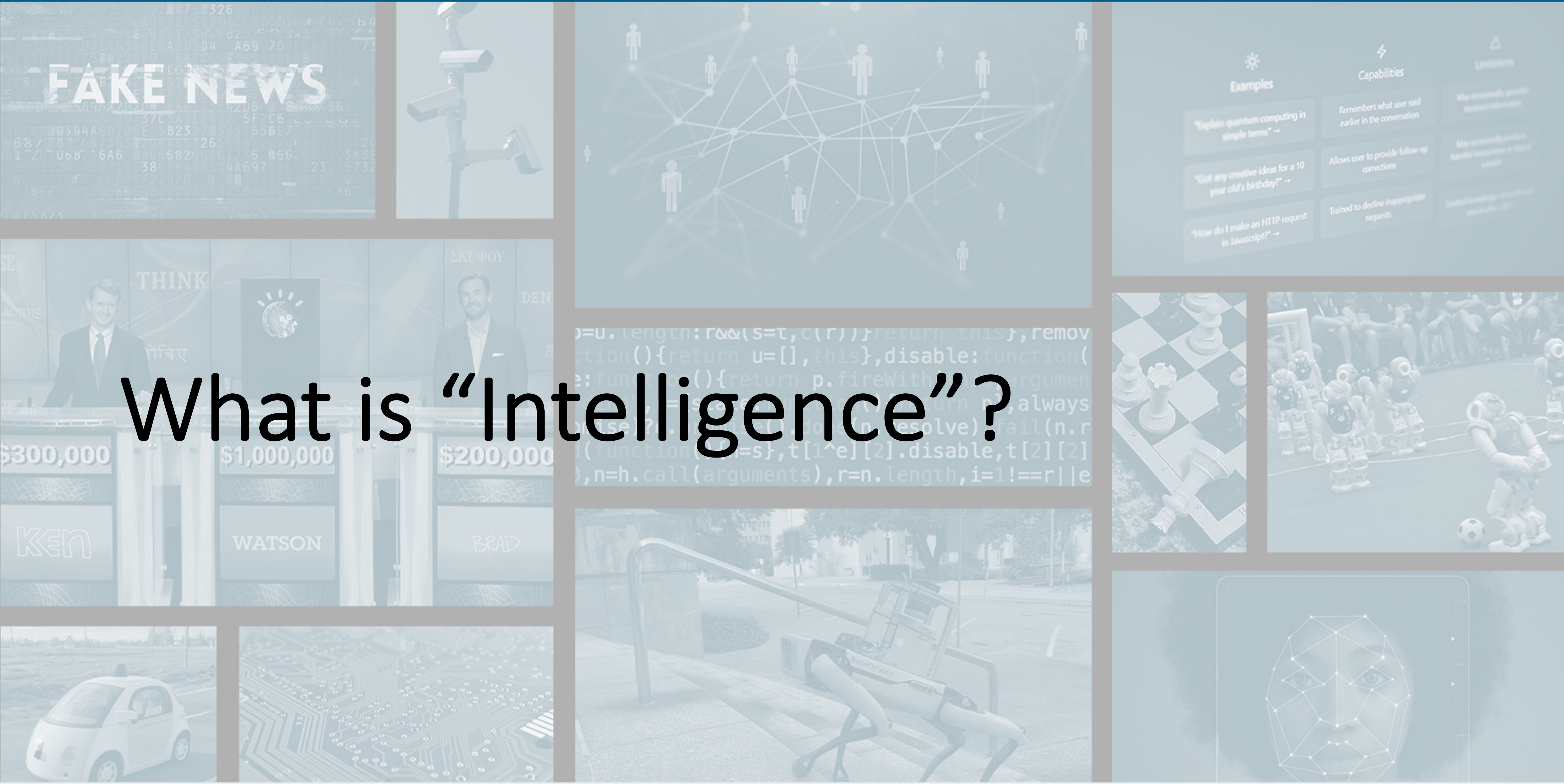
What is “Intelligence”?



https://en.wikipedia.org/wiki/Chess#/media/File:ChessSet.jpg

The Planning Problem *Applied To Chess*

Given:

1. An ***initial state*** of the world ***=> The current state of the board***
2. A set of ***available actions***, their requirements, and their effects ***=> Valid chess moves***
3. A ***goal state*** ***=> Checkmate conditions***

Compute:

A ***valid sequence of actions*** that starts from the initial state and terminates at the goal state with fewest actions ***=> Plan = sequence of chess moves***

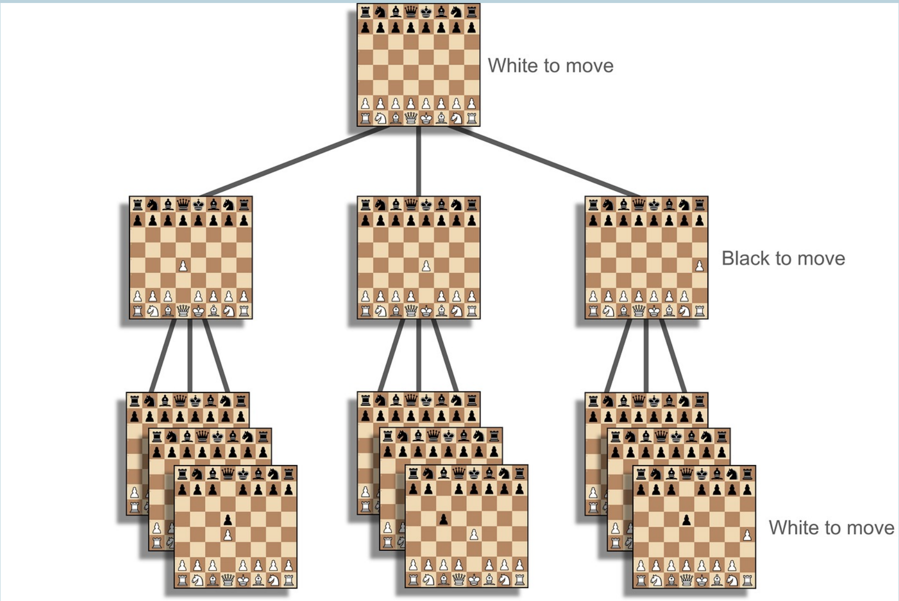
Planning Via Search (Revision)

Starting from initial state:

1. Enumerate all possible actions available, and the resulting states
2. Check if goal state reached
3. If not, for every possible outcome, repeat step 1 for all new states

Adversarial Planning Via Search

1. Enumerate all possible actions available to you
2. Enumerate all possible actions available to opponent
3. Repeat 1, 2 until game ends
4. Rewind from game end back to now:
   1. For each level pick action most favorable to the player (most games won)
   2. Stop when rewound back to now
5. Pick most favorable action (most games won)



https://blog.devgenius.io/level-up-as-a-software-engineer-by-writing-a-chess-engine-f4532f509b56

Adversarial Planning Via Search

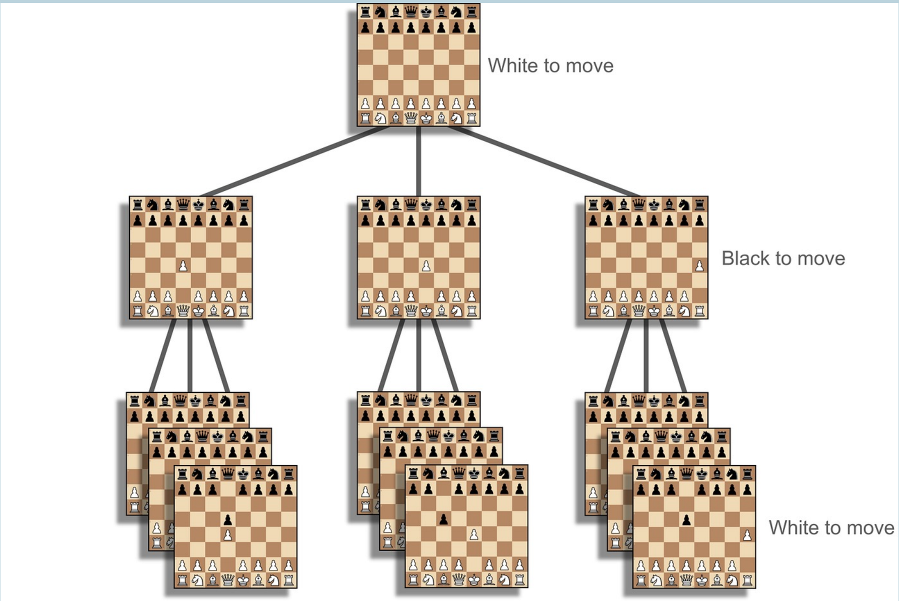
1. Enumerate all possible actions available to you
2. Enumerate all possible actions available to opponent
3. Repeat 1, 2 until game ends
4. Rewind from game end back to now:
    1. For each level pick action most favorable to the player (most games won)
    2. Stop when rewound back to now
5. Pick most favorable action (most games won)

Problem: There are too many possible game outcomes, exploring them all will take too long (and exhaust computer memory)

https://blog.devgenius.io/level-up-as-a-software-engineer-by-writing-a-chess-engine-f4532f509b56

Adversarial Planning Via Search ***+Heuristics***

1. Enumerate all possible actions available to you
2. Enumerate all possible actions available to opponent
3. Repeat 1, 2 until game ends ***up to max d steps***
4. Rewind from game end back to now:
    1. If not game end, ***use heuristic value of game state***
    2. For each level pick action most favorable to the player (***best value for player***)
    3. Stop when rewound back to now
5. Pick most favorable action (most games won)

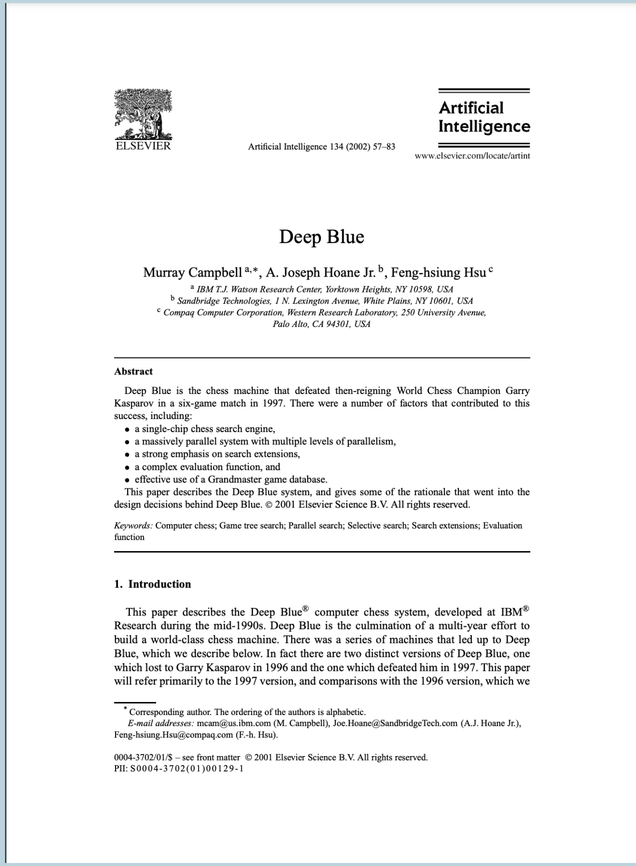


https://blog.devgenius.io/level-up-as-a-software-engineer-by-writing-a-chess-engine-f4532f509b56

1997: Deep Blue defeats Garry Kasparov

Algorithm: Tree search with some clever strategies to prune the search tree

"Secret Sauce": Special chips designed to speed up enumeration and evaluation of chess moves for tree search



ELSEVIER

Artificial Intelligence 134 (2002) 57–83

Artificial Intelligence

www.elsevier.com/locate/artint

Deep Blue

Murray Campbell [a,*], A. Joseph Hoane Jr. [b], Feng-hsiung Hsu [c]

[a] *IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA*
[b] *Sandbridge Technologies, 1 N. Lexington Avenue, White Plains, NY 10601, USA*
[c] *Compaq Computer Corporation, Western Research Laboratory, 250 University Avenue, Palo Alto, CA 94301, USA*

Abstract

Deep Blue is the chess machine that defeated then-reigning World Chess Champion Garry Kasparov in a six-game match in 1997. There were a number of factors that contributed to this success, including:

- a single-chip chess search engine,
- a massively parallel system with multiple levels of parallelism,
- a strong emphasis on search extensions,
- a complex evaluation function, and
- effective use of a Grandmaster game database.

This paper describes the Deep Blue system, and gives some of the rationale that went into the design decisions behind Deep Blue. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Computer chess; Game tree search; Parallel search; Selective search; Search extensions; Evaluation function

1. Introduction

This paper describes the Deep Blue® computer chess system, developed at IBM® Research during the mid-1990s. Deep Blue is the culmination of a multi-year effort to build a world-class chess machine. There was a series of machines that led up to Deep Blue, which we describe below. In fact there are two distinct versions of Deep Blue, one which lost to Garry Kasparov in 1996 and the one which defeated him in 1997. This paper will refer primarily to the 1997 version, and comparisons with the 1996 version, which we

[*] Corresponding author. The ordering of the authors is alphabetic.

E-mail addresses: mcam@us.ibm.com (M. Campbell), Joe.Hoane@SandbridgeTech.com (A.J. Hoane Jr.), Feng-hsiung.Hsu@compaq.com (F.-h. Hsu).

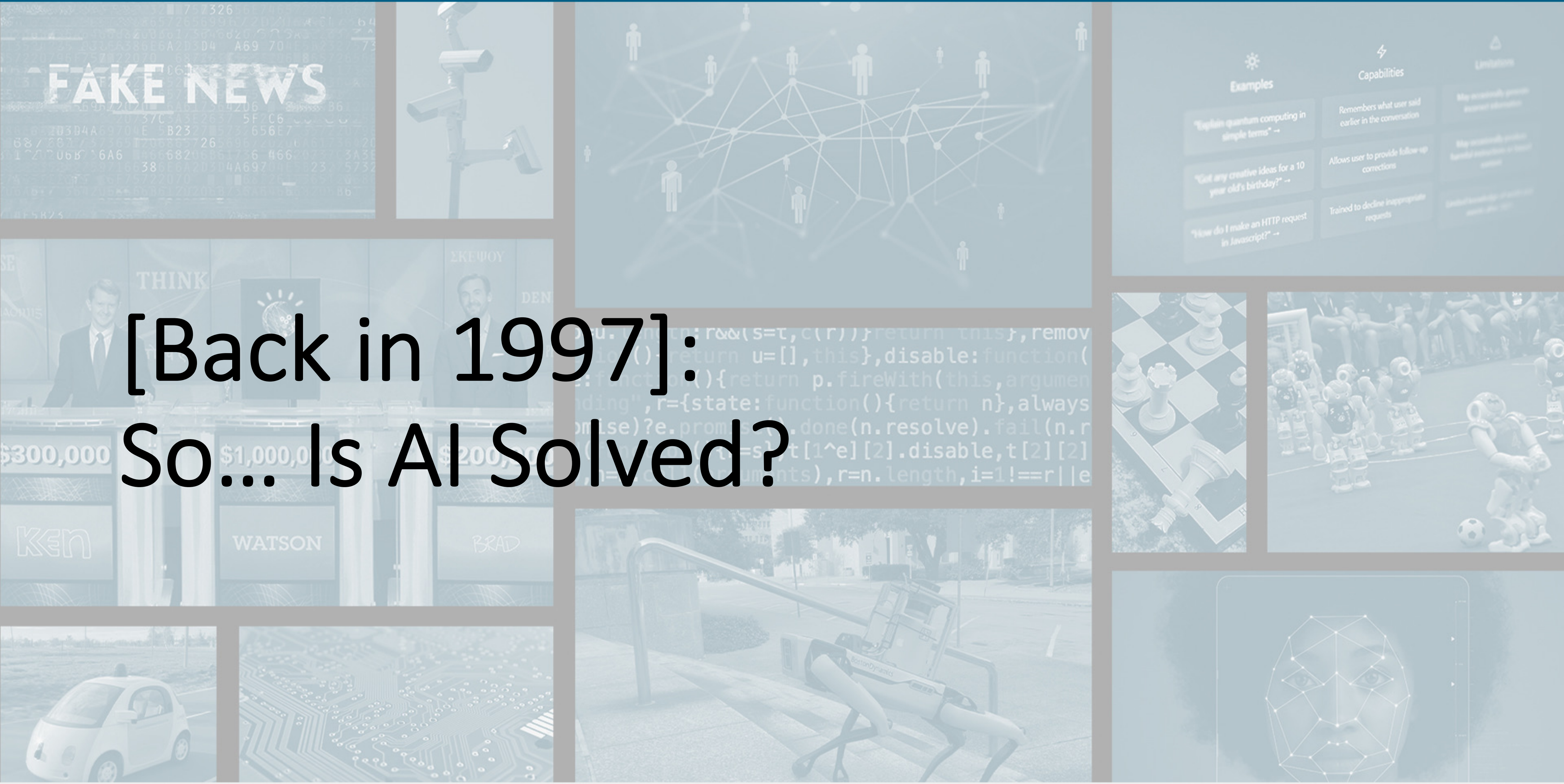
0004-3702/01/$ – see front matter

PII: S0004-3702(01)00129-1

Campbell, Murray, A. Joseph Hoane Jr, and Feng-hsiung Hsu. "Deep blue." *Artificial intelligence* 134.1-2 (2002): 57-83.

And on 8/21/2023…

https://twitter.com/IJCAIconf/status/1693798967920746711



[Back in 1997]: So… Is AI Solved?

Defining Artificial Intelligence

- A science and a set of computational technologies that are inspired by, but typically operate quite differently from, the ways people use their nervous systems and bodies to sense, learn, reason, and take action

- NOT one thing
  - More than just deep learning
  - RL, NLP, vision, planning, symbolic reasoning, algorithmic game theory, computational social choice, human computation

- Getting Computers to do the things they can't do yet
  - Once it works, it's engineering

Adversarial Planning Via Search +Heuristics

1. Enumerate all possible actions available to you
2. Enumerate all possible actions available to opponent
3. Repeat 1, 2 until game ends up to max d steps
4. Rewind from game end back to now:
    1. If not game end, use heuristic value of game state
    2. For each level pick action most favorable to the player (best value for player)
    3. Stop when rewound back to now
5. Pick most favorable action (most games won)

1. **There are games (e.g., Go) where 1,2 are intractable even for a few steps.**
2. **We don't want to create special-purpose computer chips _for every specific problem_**

Outside of Planning World, In 2012…

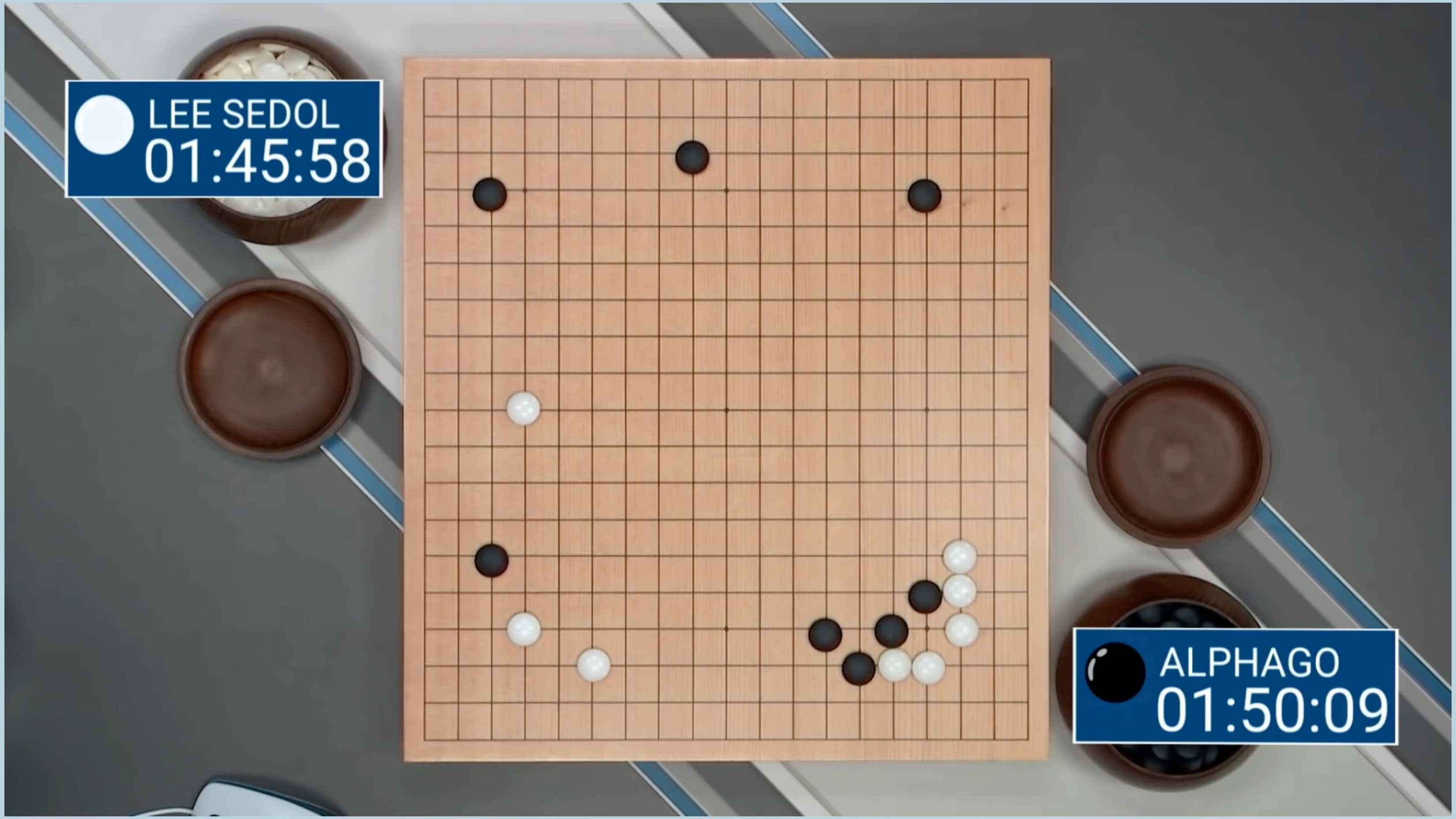
In the computer vision community, deep neural networks gain steam…

[Story for another lecture: Computer Vision]

2016: AlphaGo Defeats Lee Sedol at Go

AlphaGo - The Movie: https://www.youtube.com/watch?v=WXuK6gekU1Y

2016: AlphaGo Defeats Lee Sedol at Go



AlphaGo - The Movie: https://www.youtube.com/watch?v=WXuK6gekU1Y

Adversarial Planning Via Tree Search *+Deep Learning*

1. Enumerate ***promising actions [policy network]*** available to you
2. Enumerate ***promising actions [policy network]*** available to opponent
3. Repeat 1, 2 until game ends up to max d steps
4. Rewind from game end back to now:
    1. If not game end, use ***learned value [value network]*** of game state
    2. For each level pick action most favorable to the player (best value for player)
    3. Stop when rewound back to now
5. Pick most favorable action (most games won)