

Martin Mundt, Dr. Iuliia Pliushch, Prof. Dr. Visvanathan Ramesh

Pattern Analysis & Machine Intelligence

Praktikum: MLPR-WS19/20

Week 1: Introduction

Lecture requirements

- 4 hours per week, 8 credit points -> attend the class!
- Final class group projects. Will start at the end of the lecture period with presentations at the end of the semester (we will choose a date together later).
- Bring a laptop with WiFi access to the university network (Flughafen, eduroam etc.).
- Each session is going to start with a roughly 30-60 minute lecture.
- Online notebooks will be used for the 3 hour long practical sessions.
- We will use Google's Colaboratory for cloud computation. If you have a Gmail, android etc. account, then it is the same account you already have.
- If you don't want to use Colab (because you don't want a google account), you can execute the notebooks locally, we advise a Linux or Mac OS (and will provide no Windows support). Please be reminded that some of the later deep learning content will be difficult to setup/run on your local laptops (unless they have a GPU) as we will use Google Colab's free GPU instances.
- **All lecture materials will be shared on GitHub:**
https://github.com/ccc-frankfurt/Practical_ML_WS19

Schedule

Introduction:

Week 1: 14.10 – Introduction, python tools review, software management (version-control & documentation)

Week 2: 21.10 – Ideas behind ML, gradient descent on functions, logistic regression -> Kaggle Titanic dataset

Block 1 - Supervised Learning:

Week 3: 28.10 – Random forests from scratch -> Revisiting Titanic

Week 4: 04.11 – Random forests application, intro to sklearn -> San Francisco crime challenge

Week 5: 11.11 – Basic neural networks from scratch -> Multi-layer perceptron for classification of fashion images

Week 6: 18.11 – Introduction to PyTorch, deep learning, convolutional neural networks -> Reading traditional Japanese characters (Kuzushiji)

Week 7: 25.11 – Neural sequence models, recurrent neural networks -> Shakespeare poetry text generation

Schedule

Block 2 - Unsupervised Learning:

Week 08: 02.12 – Unsupervised learning: k-means clustering and principal component analysis -> self-generated known data distributions

Week 09: 09.12 – Unsupervised neural networks, autoencoders (representation learning/unsupervised pre-training) -> Revisiting fashion and Kuzushiji images

Week 10: 16.12 – Generative models: variational autoencoders -> handwriting generation

Week 11: 13.01 – Generative models: generative adversarial networks -> Face generation

Block 3 – Reinforcement Learning

Week 12: 20.01 – Classic Q-learning -> Q-learning

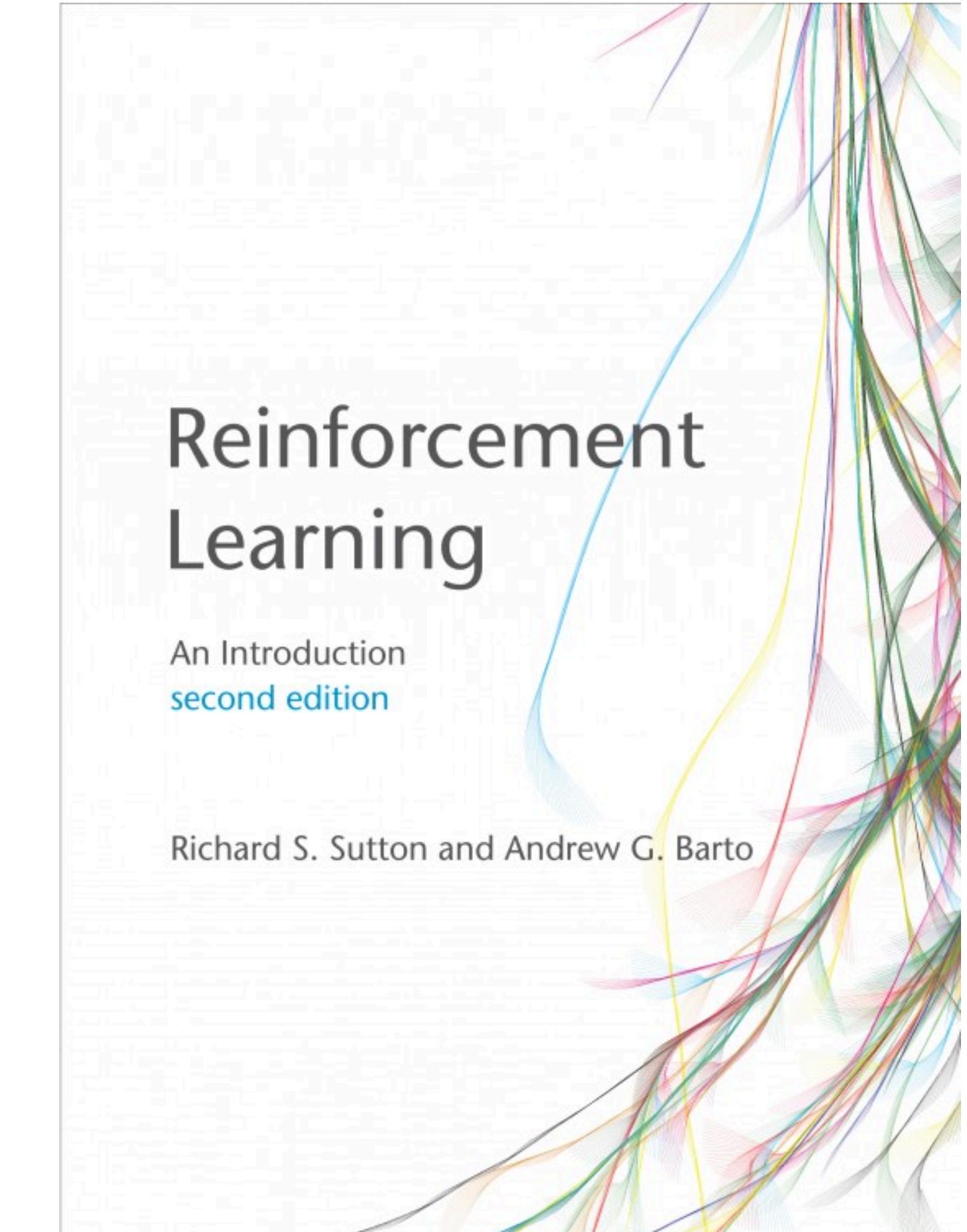
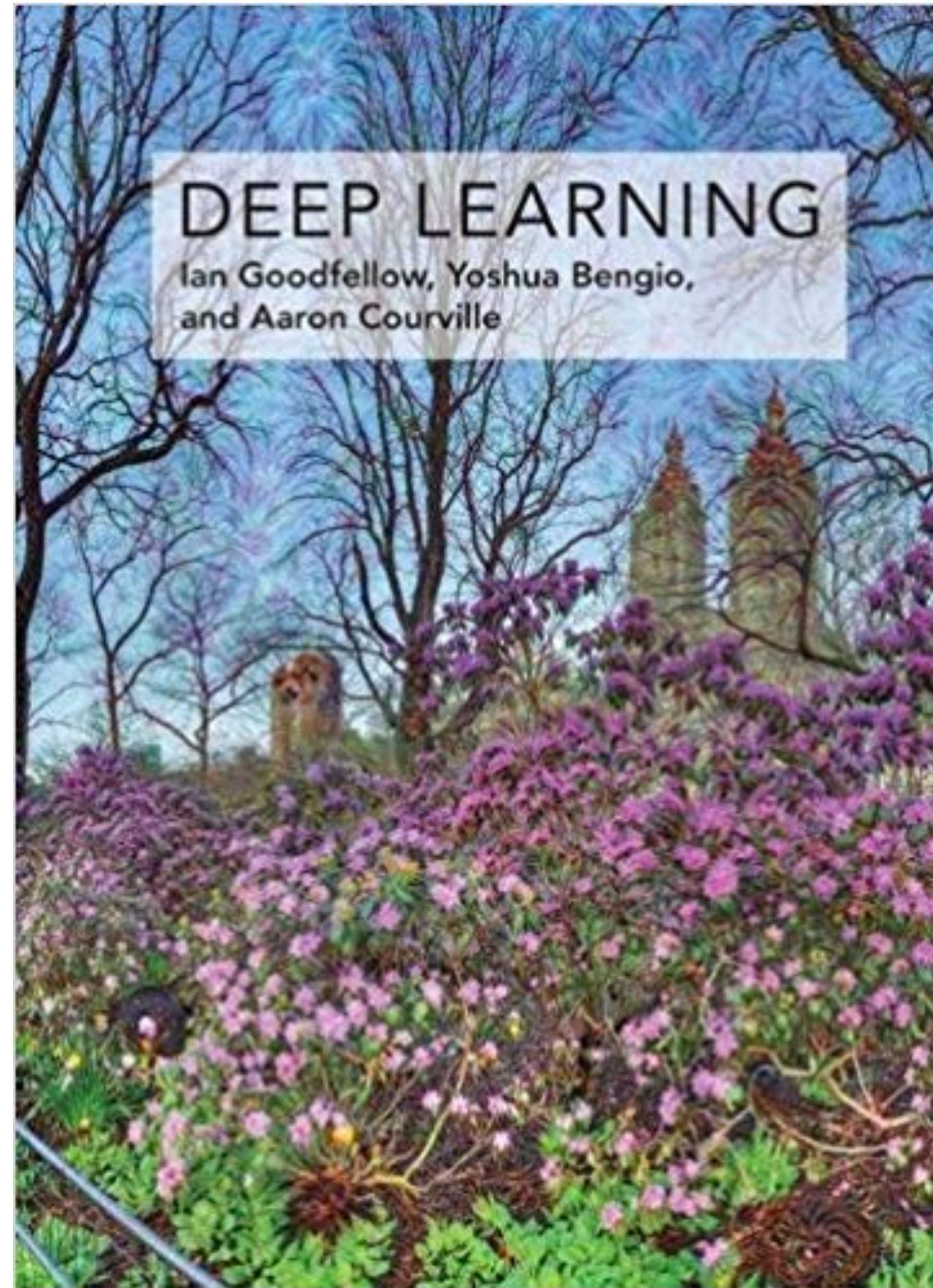
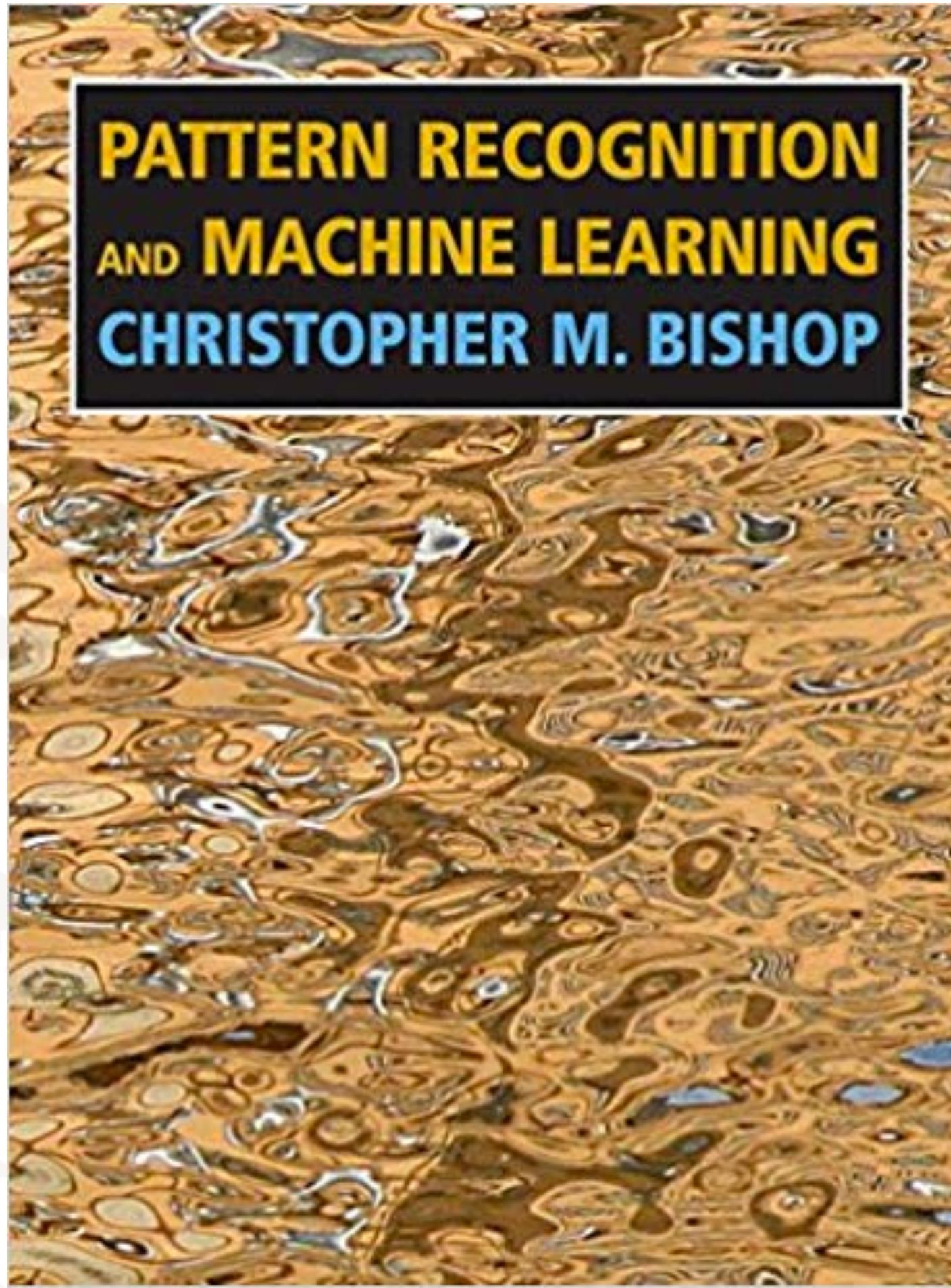
Week 13: 27.01 – Deep reinforcement learning, QNN -> Taxi driver

Week 14: 03.02 – Reinforce algorithm -> Robotic application (walking/grasping)

Project planning:

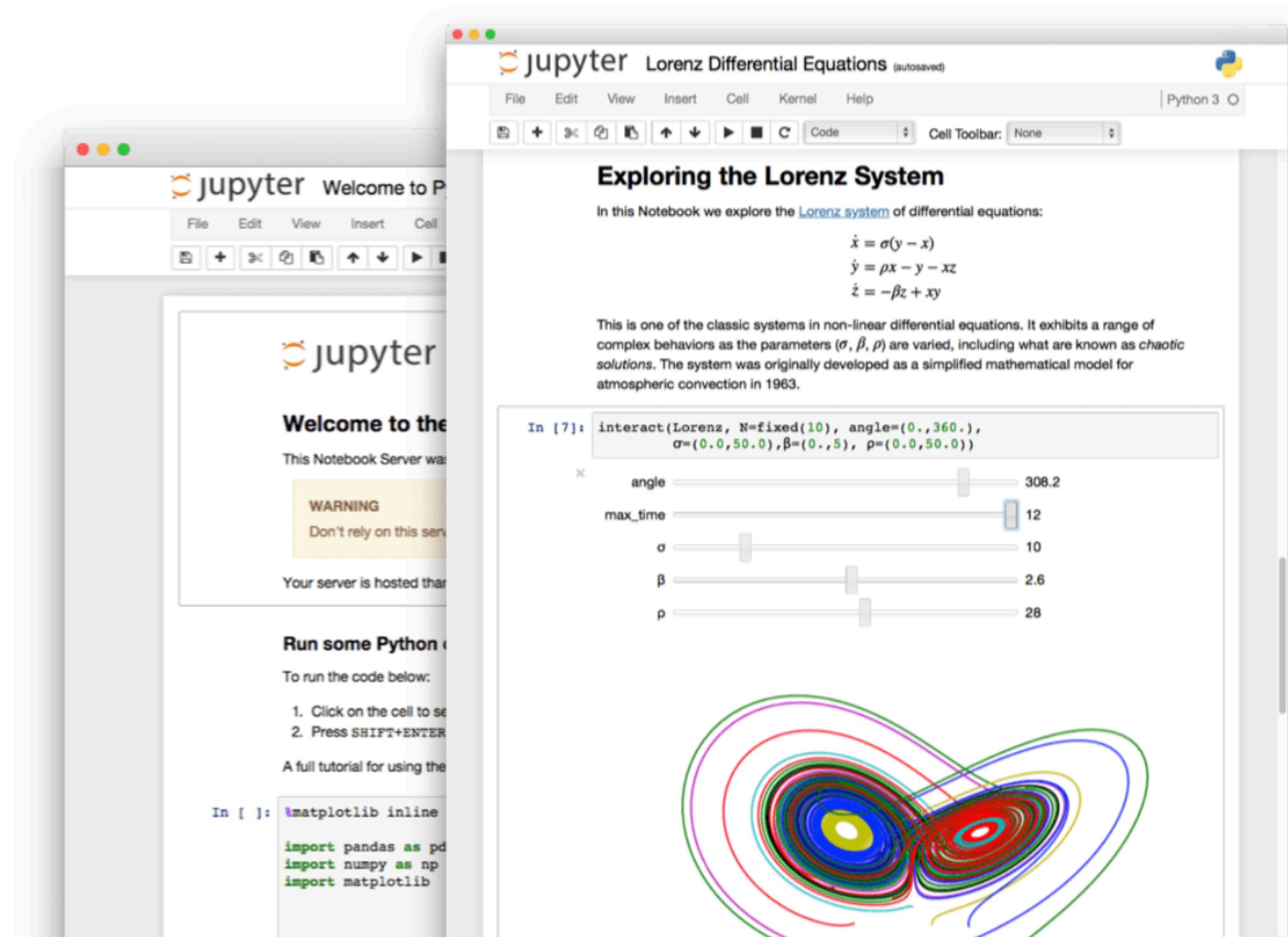
Week 15: 10.02 – State-of-the-art, open questions and existing issues -> Project pitches (3 slides per person)

Literature



Jupyter notebooks: <https://jupyter.org/>

- Execution of python code in cells
- We will provide step-by-step notebooks where sections are missing and need to be filled in



The Jupyter Notebook interface is shown, featuring a main notebook window and a sidebar. The notebook displays code for generating a Lorenz attractor plot, along with mathematical equations and parameter sliders. The sidebar shows a 'Welcome to the Jupyter Notebook' message.

The Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

[Try it in your browser](#) [Install the Notebook](#)

Install About Us Community Documentation NBViewer JupyterHub Widgets Blog

Google Colab: <https://colab.research.google.com/>

- Google Colab allows execution of notebooks in the cloud from a browser
- GPU/TPU instances are free with an execution time-out after 1 hour (enough for our purposes)

The screenshot shows the Google Colab interface with the title "Overview of Colaboratory Features". The left sidebar contains a "Table of contents" with links to various sections like "Code cells", "Text cells", "Adding and moving cells", etc. The main content area is expanded to show the "Cells" section, which includes a sub-section on "Code cells". It describes how to run code cells using various methods and mentions the "Runtime" menu for additional options. A preview window at the bottom shows a code cell with the following content:

```
[ ] 1 a = 10  
2 a
```

10

Machine learning in our lives: some examples

Medicine/Pharma

Finance

Smart homes

Video games

Customer support, chat bots

Machine translation

Smart cars

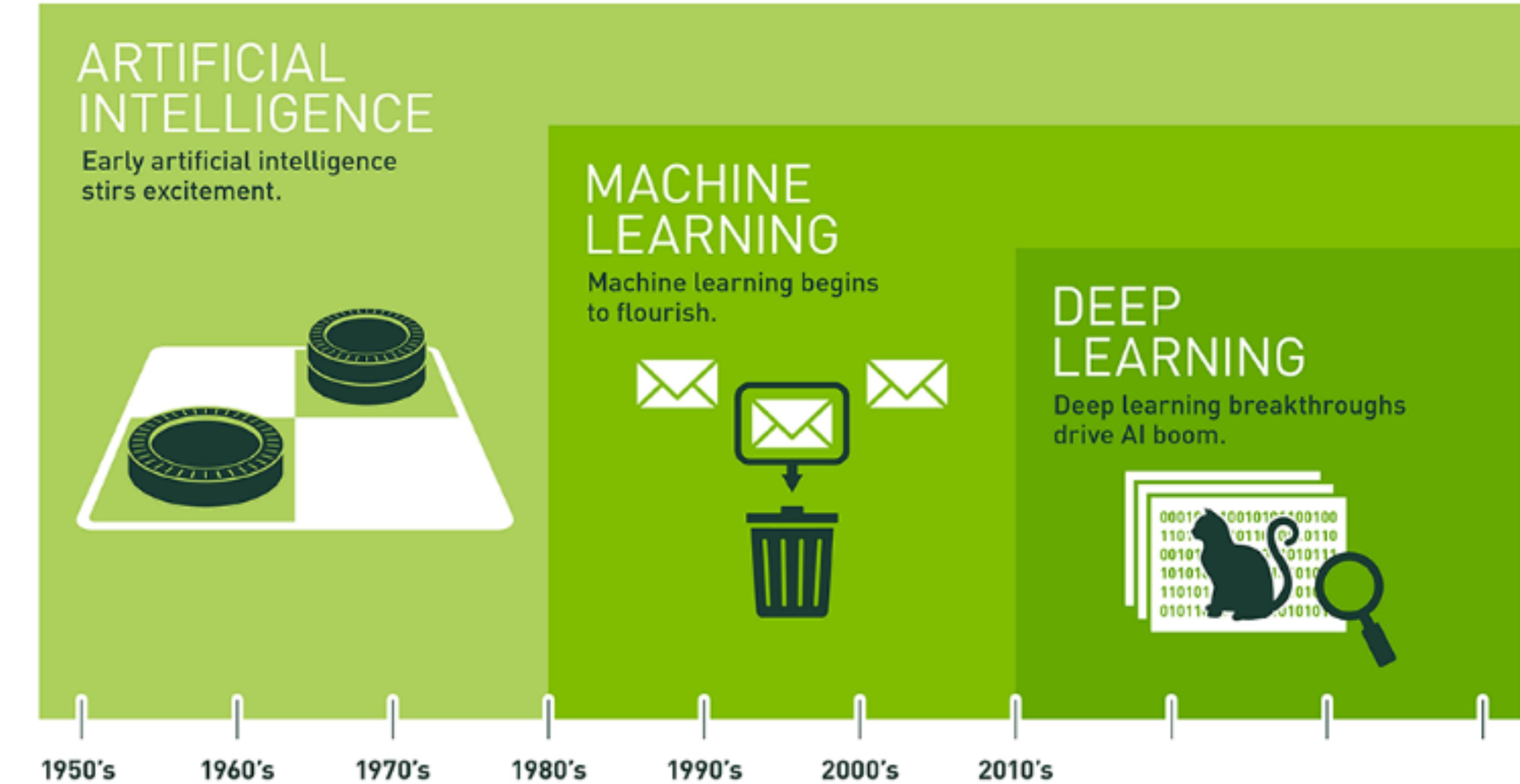
Surveillance, security

Recruiting, job market

Robot control

Art

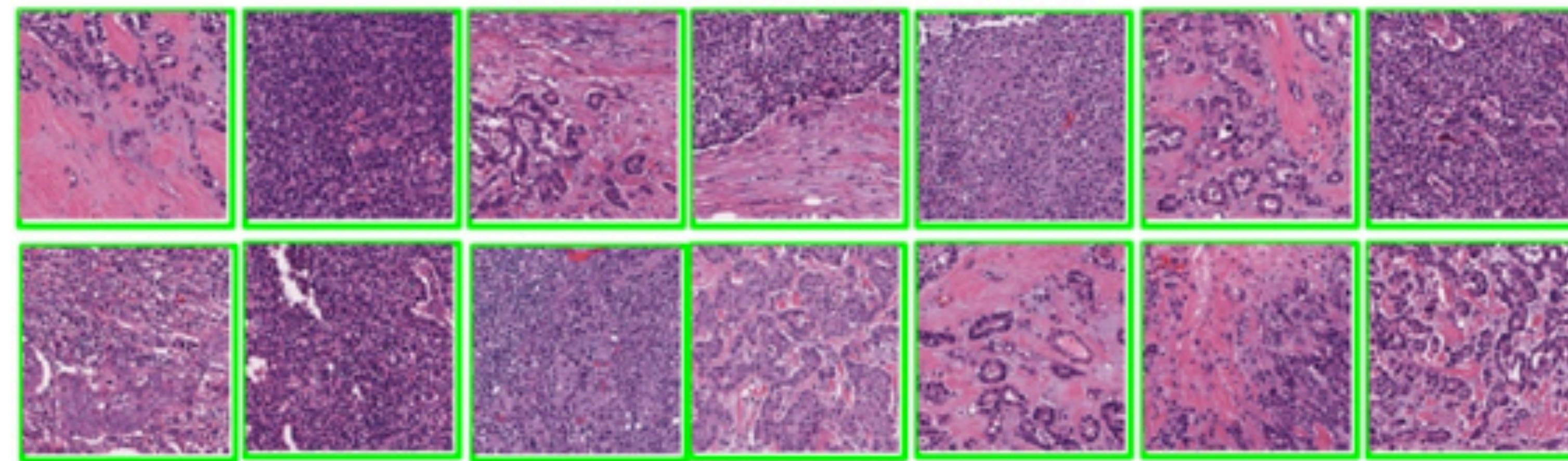
Terminology



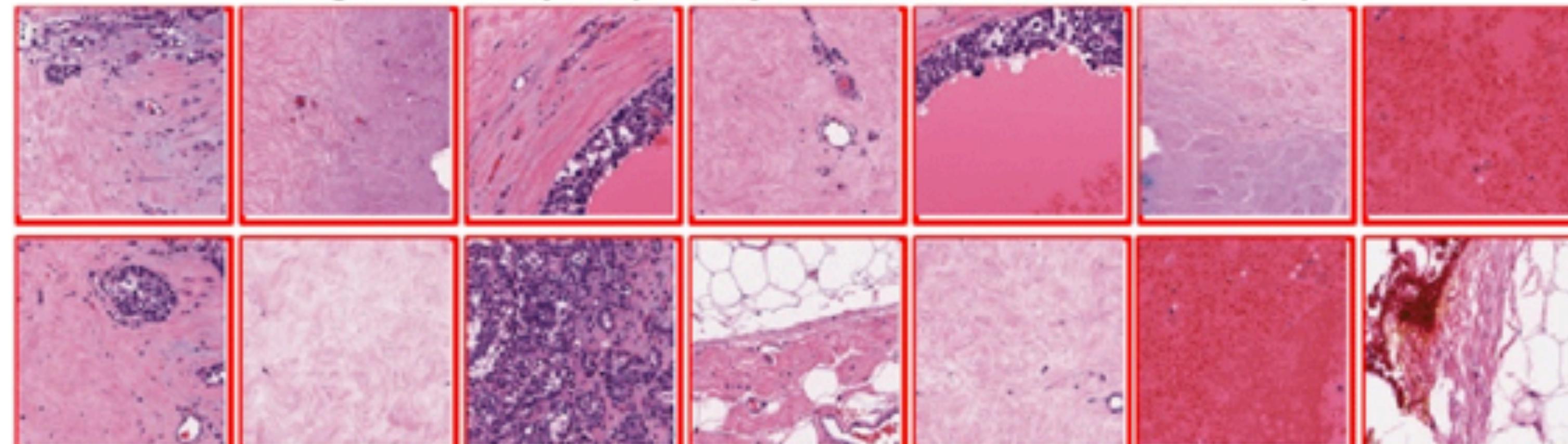
Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

Medical Diagnosis

Positive examples (IDC tissues)



Negative examples (healthy or not invasive tumor tissues)



<https://doi.org/10.1117/12.2043872>

Personal Assistants – Natural Language Processing

The image displays two screenshots of the Google Assistant app interface.

Screenshot 1 (Left): The title bar reads "Google Assistant Preview Edition". The time is shown as "TODAY • 3:59 PM". A "News" button is visible in the top right. The assistant says "Here are the top stories" and shows two news cards:

- A thermometer image with the text "Row as airline weighs passengers to American Samoa" and "14 hours ago - www.bbc.co.uk".
- A close-up of a computer screen displaying the "YAHOO!" logo.

The assistant asks "Ok. What time would you like me to send it? You can type any time you want." Below are six time buttons: "Cancel", "7am", "9am", "11am", "12:30pm", and "3pm". At the bottom is a text input field with "+ Say something..." and a microphone icon.

Screenshot 2 (Right): The title bar reads "Google Assistant Preview Edition". The user asks "What's the weather like there?". The assistant replies "It looks pleasant outside" and shows the weather for San Francisco: "65°F in San Francisco" with "Partly Cloudy - High: 66° Low: 54°". A sun and clouds icon is shown. Below is another text input field with "+ Say something..." and a microphone icon.

Smart Home



<http://blog.airpatrol.eu/wp-content/uploads/2016/10/bigstock-Smart-Home-103718372-1200x800.jpg>

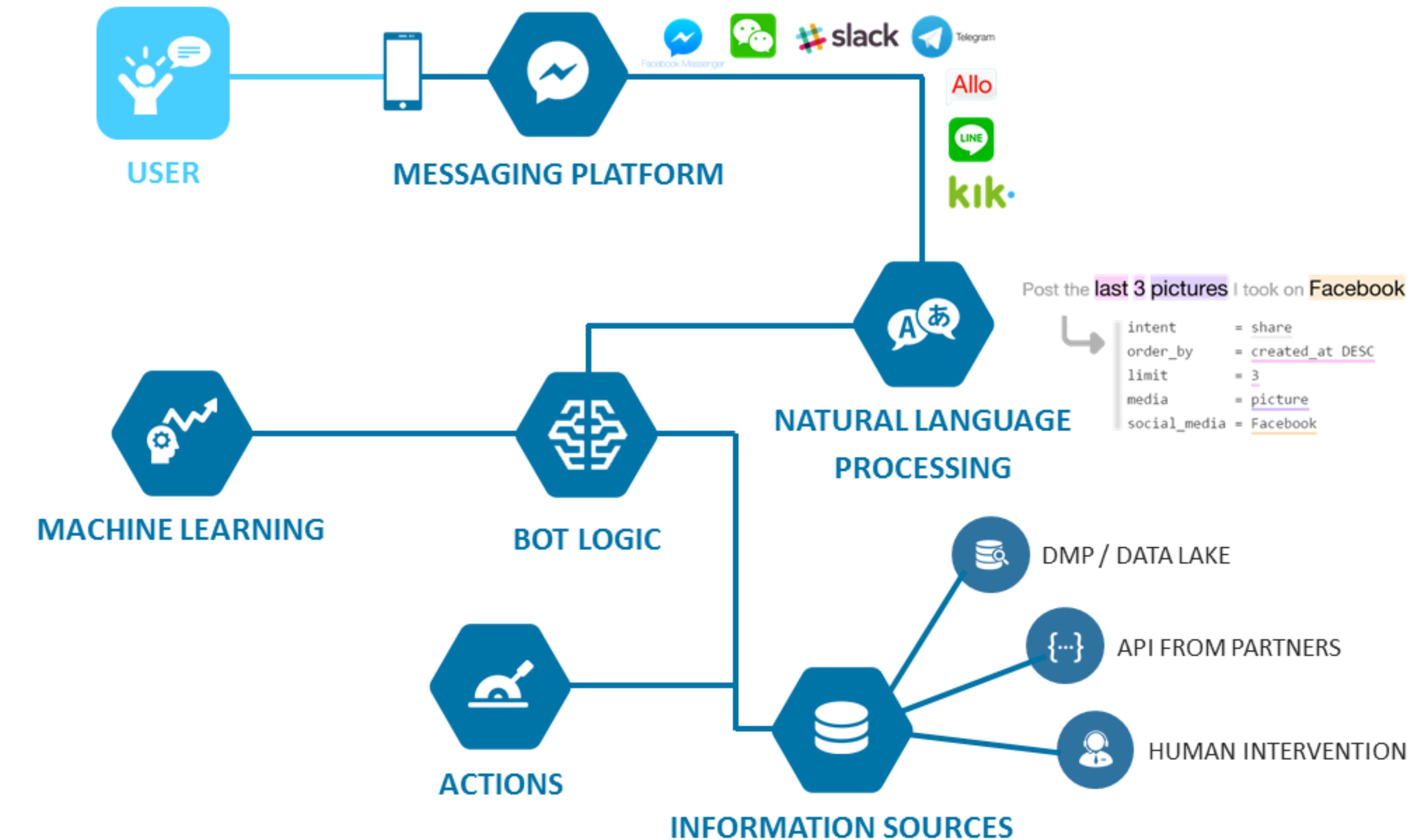
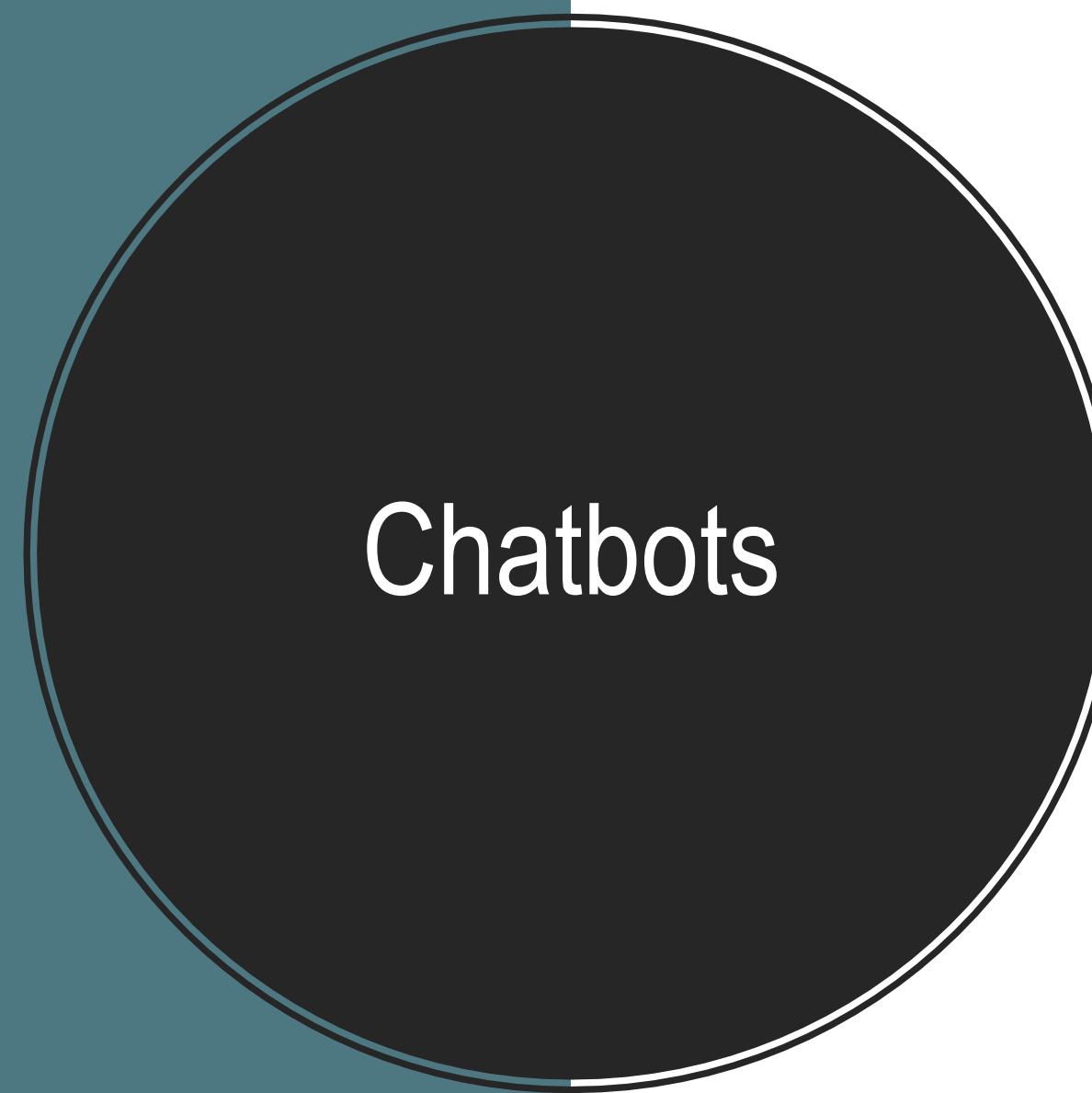


Video Games

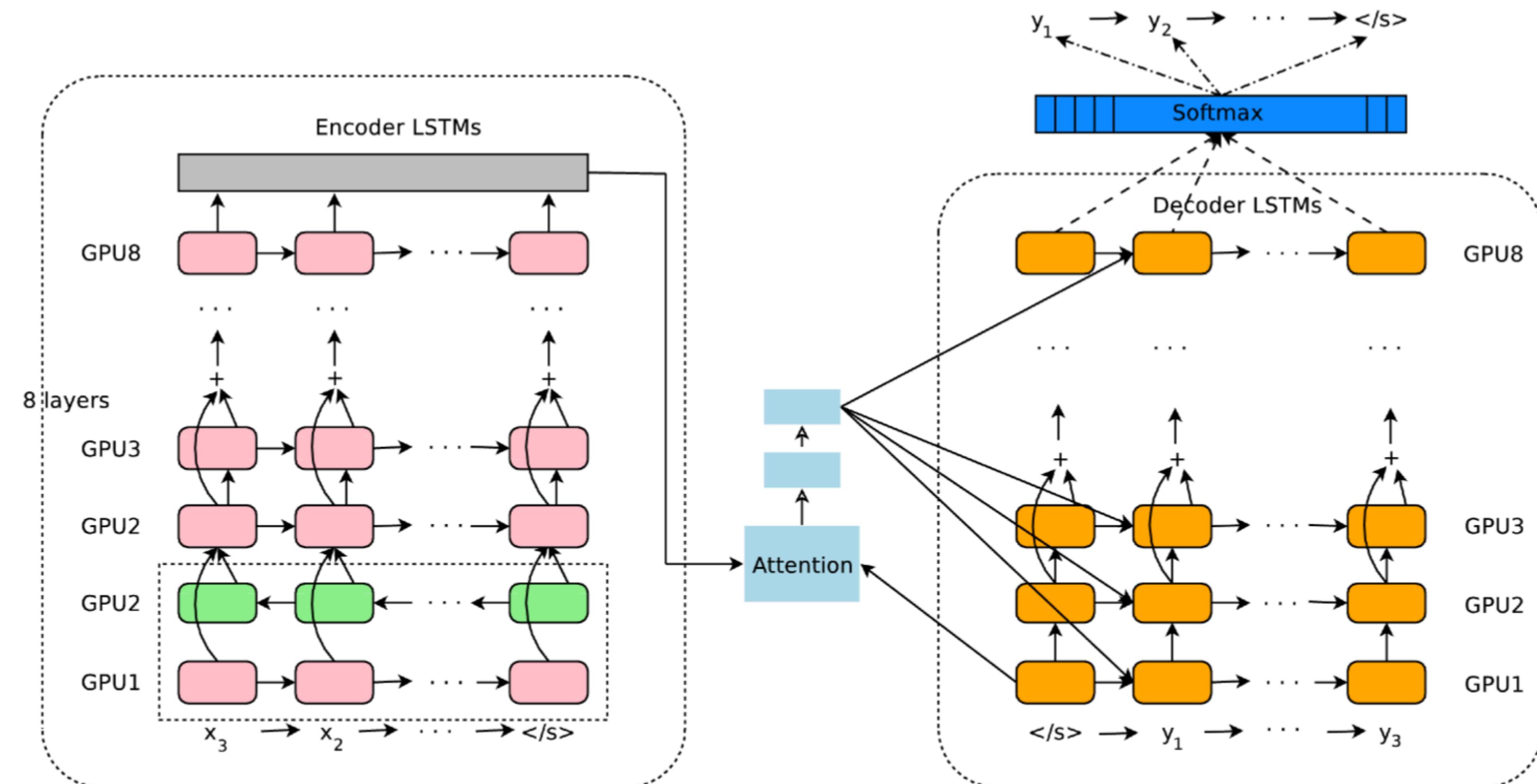
DEBUG MODE:

SHOW THE AI PERFORMS THE SEARCH.

Video at this link <https://www.youtube.com/watch?v=36cKiPPrGho>



Machine Translation



[From Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation]

Autonomous driving

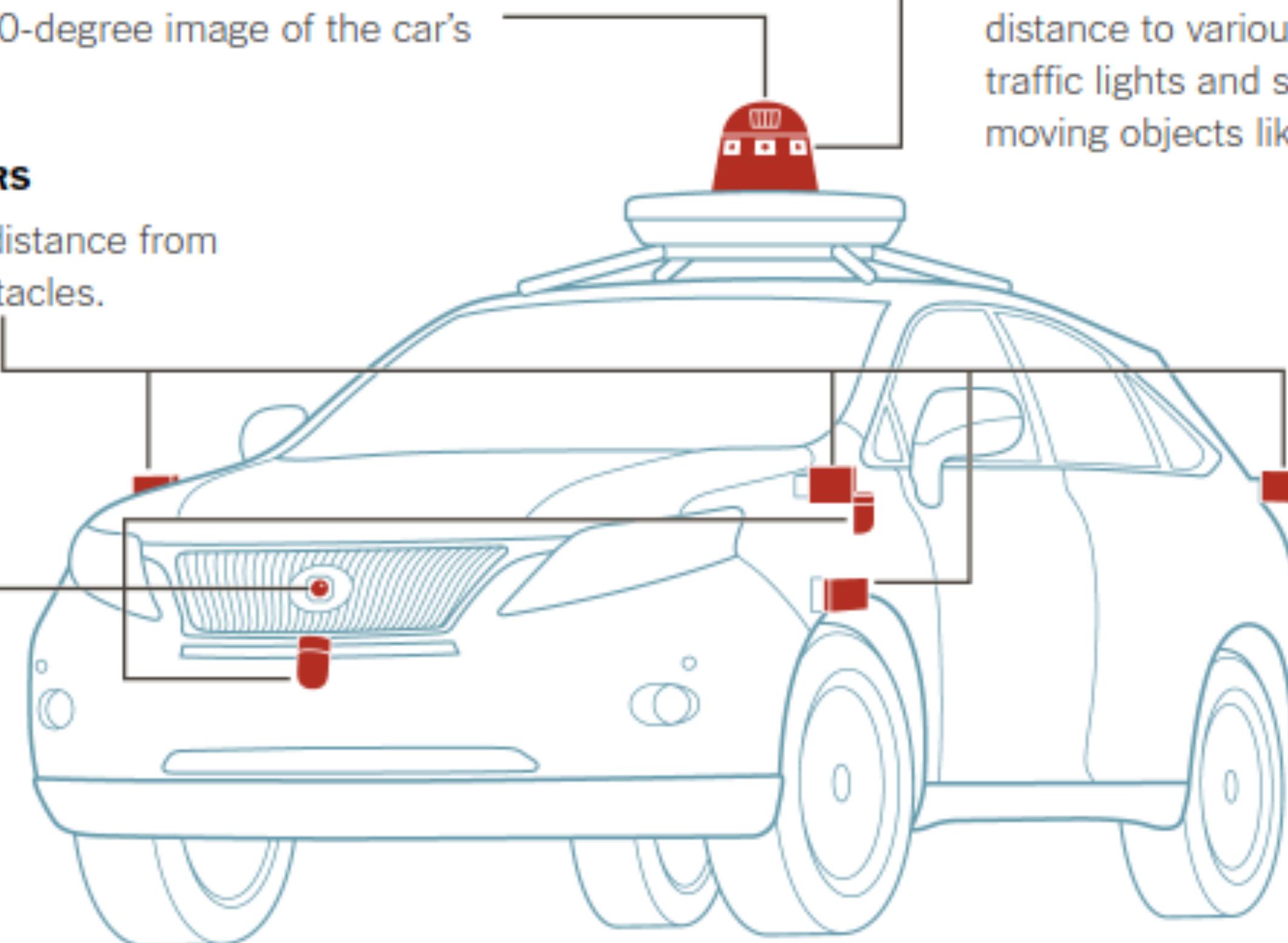
LIDAR UNIT

Constantly spinning, it uses laser beams to generate a 360-degree image of the car's surroundings.

RADAR SENSORS

Measure the distance from the car to obstacles.

ADDITIONAL LIDAR UNITS



CAMERAS

Uses parallax from multiple images to find the distance to various objects. Cameras also detect traffic lights and signs, and help recognize moving objects like pedestrians and bicyclists.

MAIN COMPUTER (LOCATED IN TRUNK)

Analyzes data from the sensors, and compares its stored maps to assess current conditions.

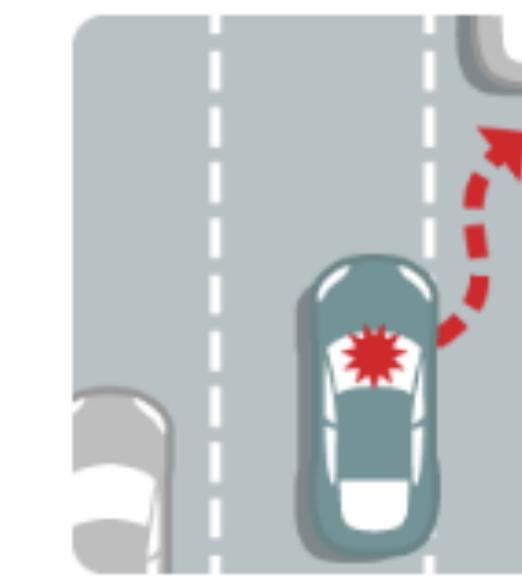
By Guilbert Gates | Source: Google | Note: Car is a Lexus model modified by Google.

Autonomous driving



Collision avoidance

Radar-, laser-, or camera-based systems warn of an impending collision, and can automatically apply the brakes in some cases.



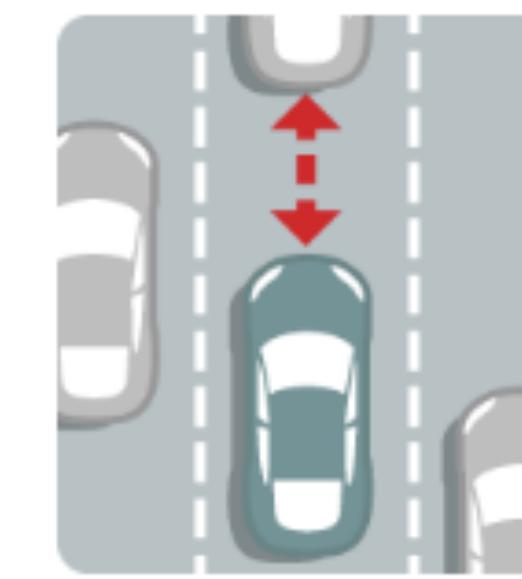
Drifting warning

If a car begins to deviate from its lane, some systems alert the driver and apply a small counter-steering force.



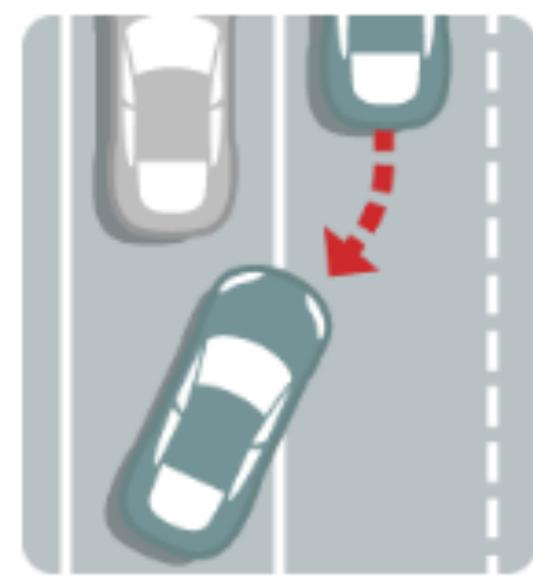
Blind-spot detectors

Cameras or radar can detect vehicles in the driver's blind spot and then alert the driver with sounds or warning lights.



Enhanced cruise control

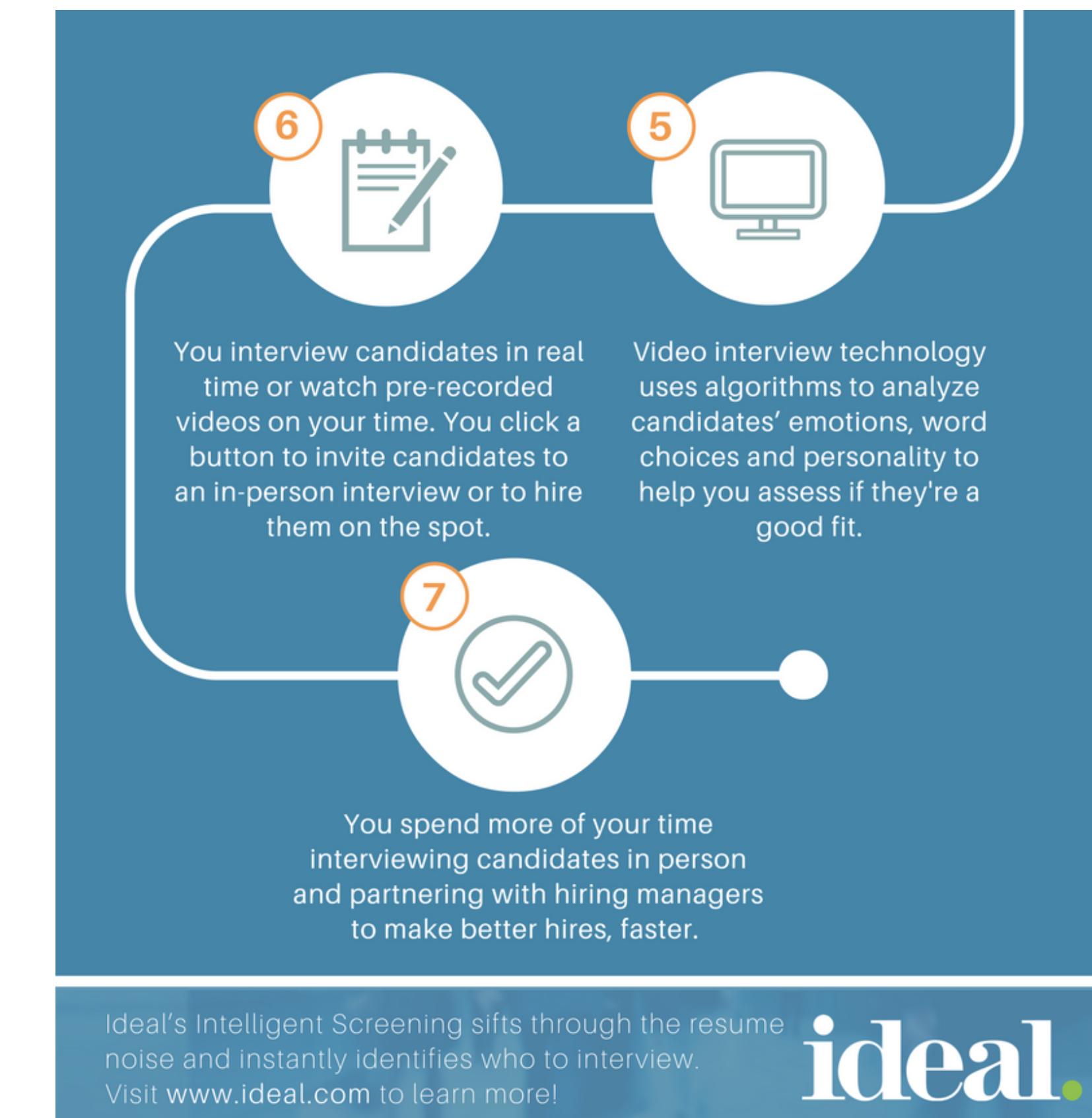
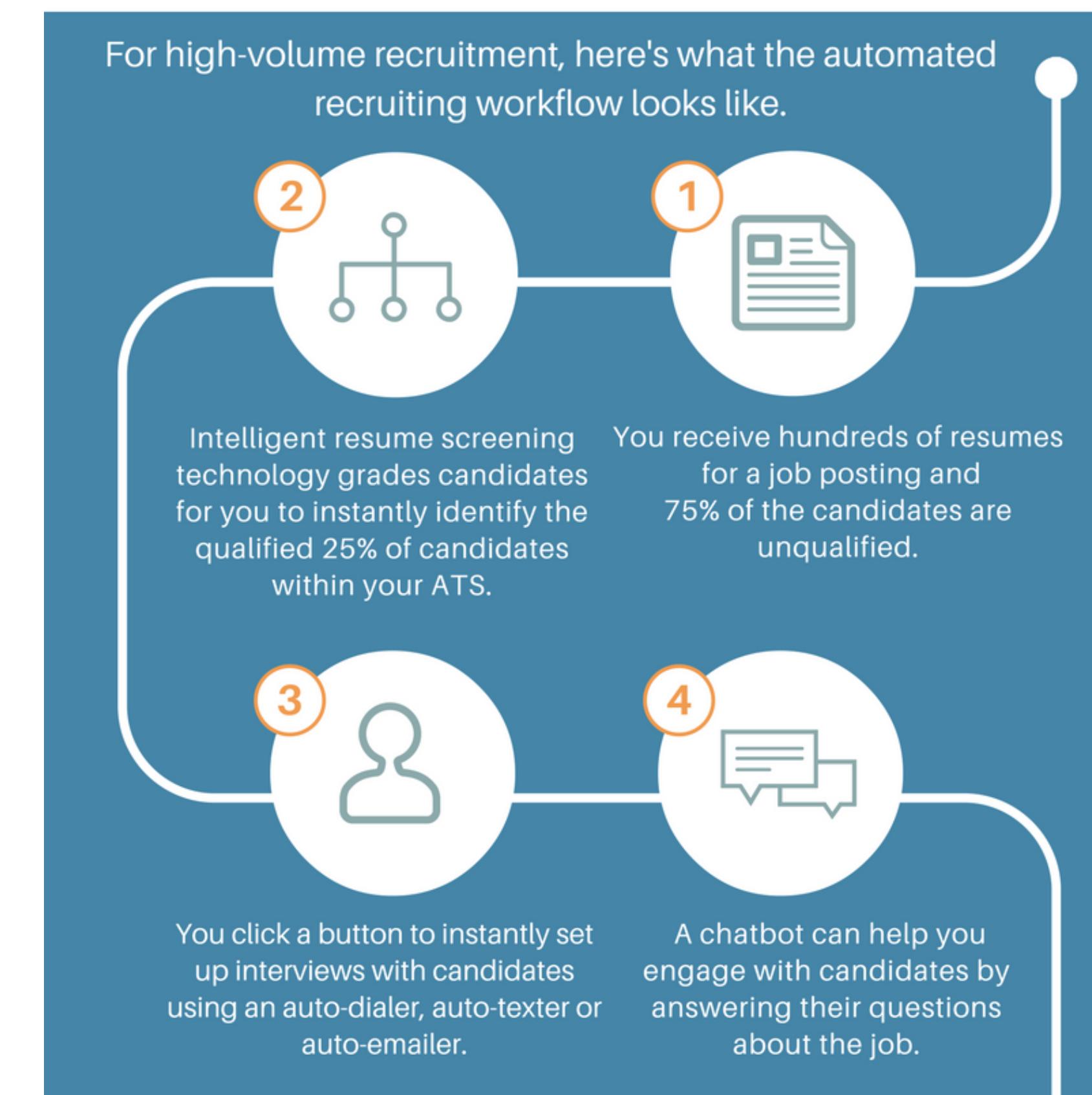
A predefined distance can be maintained to the vehicle ahead. If it slows, your car also slows.



Self parking

The car maneuvers itself into a parking spot using cameras or sonar. But the driver usually has to brake and follow commands.

Recruiting



ideal.



Art

1

Upload photo

The first picture defines the scene you would like to have painted.



2

Choose style

Choose among predefined styles or upload your own style image.



3

Submit

Our servers paint the image for you. You get an email when it's done.



<https://deepart.io/>

Sketching



https://magenta.tensorflow.org/assets/sketch_rnn_demo/img/multi_sketch_mosquito.gif

Photography: night mode



<https://www.fastcompany.com/90247454/the-pixel-3-puts-googles-extraordinary-ai-in-your-pocket>