Intro. To
Machine Learning &
Computer Vision

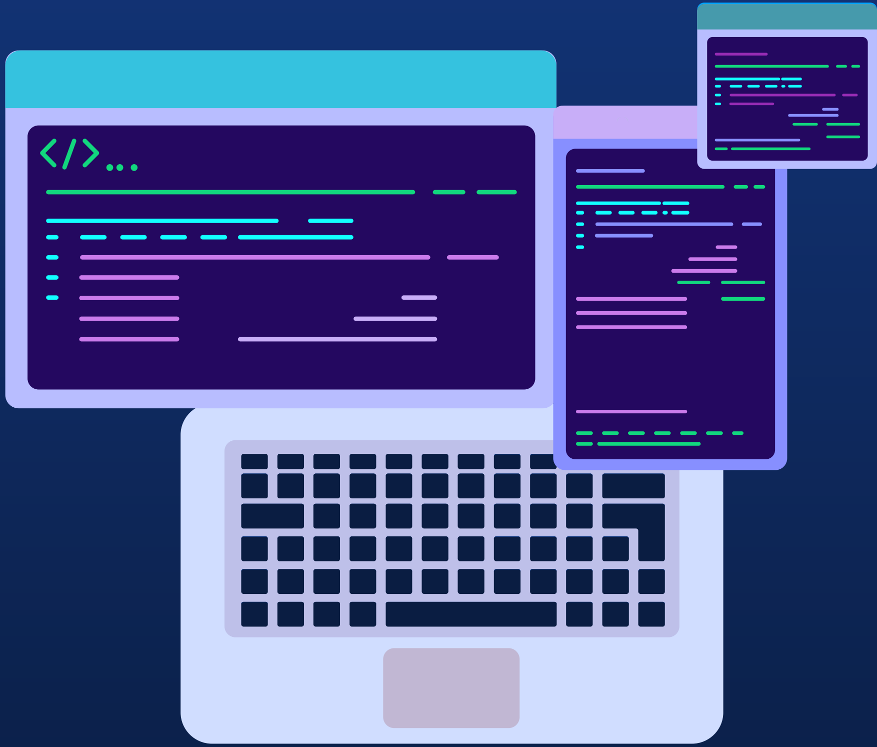


TABLE OF CONTENTS

01

Intro to Machine Learning

02

Introduction to Computer Vision

03

Using RL!

What is BigThink?

BigThink AI’s goal is to bring awareness towards the growing field of artificial intelligence. We intend to advance our members’ knowledge and experience with fields within and related to AI through meetings, discussions, seminars, and projects.

What our Club Does?

Pyoneers

Weekly “lectures” where our leads do a brief lecture followed by an example

Edge

A group of students with machine learning research experience work together on a semester long research project.

01

Intro to Machine Learning

What is Machine Learning?

- Machine learning is a subset field of artificial intelligence
- Utilizes mathematics and statistics to learn from data
- Before ML, we write programs with algorithms that have deterministic behavior
  - Ideal for scenarios with complex rule sets that are not deterministic
- Aims to generalize the patterns in the data
- Two Main Problems
  - Classification and Regression
- We start with training data that can be labeled or unlabeled.
- Then we feed into a machine learning model, evaluate, and then deploy

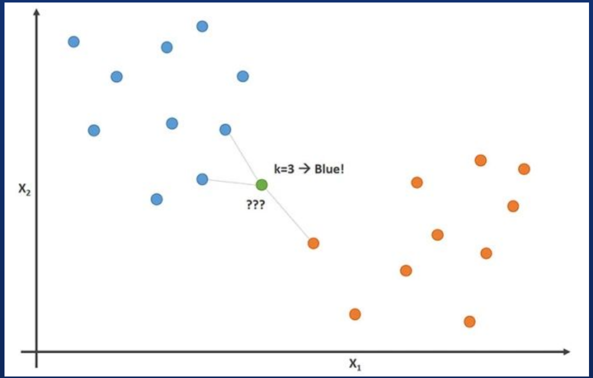
Ex (Spam detection): It is difficult to write all the rules to detect spam in emails. Machine learning can adapt to changes over time.

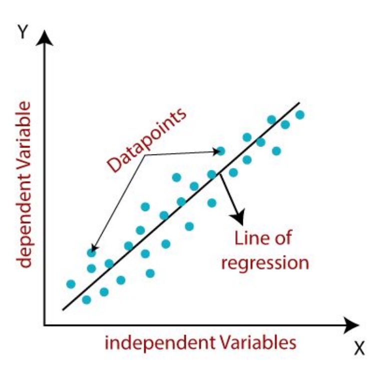
Supervised Learning

- Supervised learning is when the data is labeled, so we have the input and the expected output
- Algorithm predicts classes on new data

Examples

- k-Nearest Neighbors
- Linear Regression
- Decision Trees and Random Forests
- Neural Networks



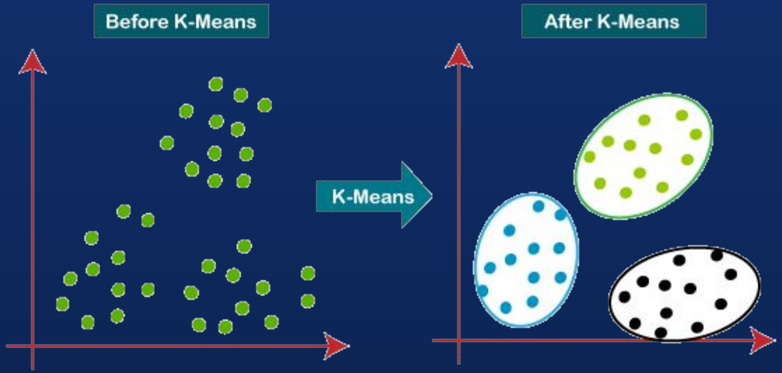


Unsupervised Learning

- The data is not labeled
- Algorithm find patterns or groups on its own without knowing how they are grouped

Examples

- Clustering
- Anomaly Detection

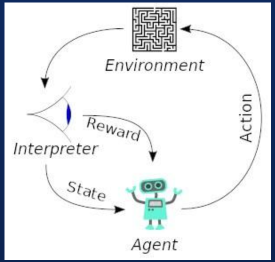


Reinforcement Learning

- Between supervised and unsupervised
- Does not rely on labeled data, but performs "desired" actions based on rewards
- Key Terminology:
  - Agent - What moves
  - State - Where the agent moves
  - Action - A move that can be taken
- The system gives a reward based on its actions, and updates what action it should take top maximize reward

Examples

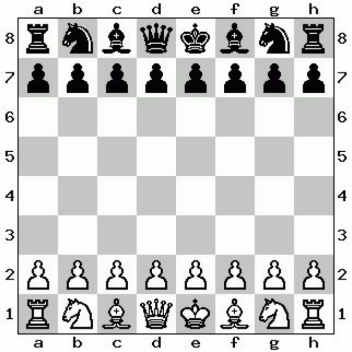
- Games!
- Self driving cars!



What are the state, action, and agent in a game of chess?

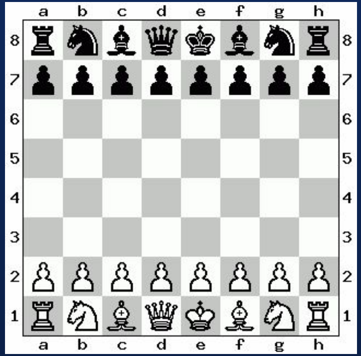
Quiz!

- Q: What are the state, action, and agent in a game of chess?



Answer!

- Q: What are the state, action, and agent in a game of chess?
  - State - The current board position
  - Actions - Any legal move
  - Agent - The person/entity making the decisions



02

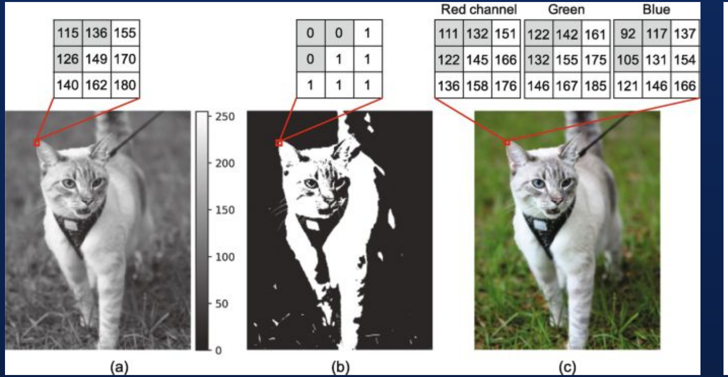
Intro to Computer Vision

What is Computer Vision (CV)?

- Computer vision detects patterns and objects in images by identify each pixel to learn patterns
- New developments in Neural Networks paved way for computer vision
- Common applications are image segmentation, classification, object tracking, and detection

How does it work?

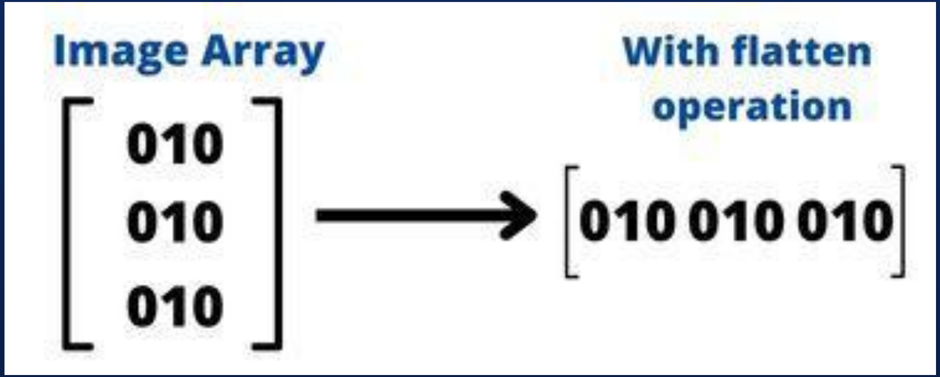
- We are given photos. The model will analyze the photos by each pixel
- Each pixel has 3 channels (for RGB) that ranges from 0-255 in red, green, and blue channels
- We then flatten the image to a single vector to feed into our model
- The model then learns to group or categorize the images based on similarities it finds in the patterns of these pixels, without being explicitly told what to look for



115	136	155
126	149	170
140	162	180

0	0	1
0	1	1
1	1	1

Red channel			Green			Blue		
111	132	151	122	142	161	92	117	137
122	145	166	132	155	175	105	131	154
136	158	176	146	167	185	121	146	166

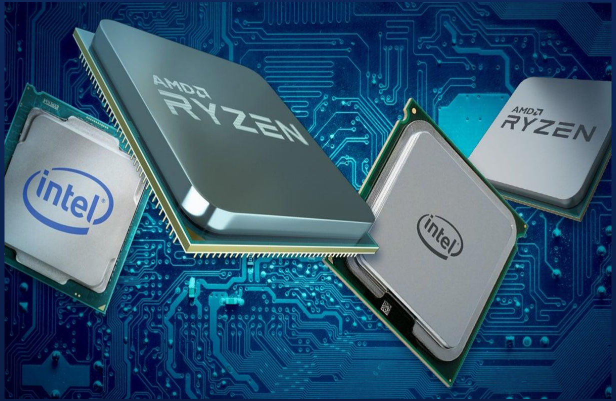


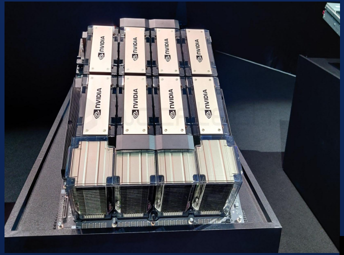
Modern Challenges in CV

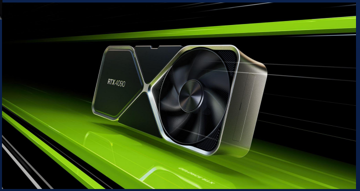
- Weak performance to train our models
- Training on large images is computationally expensive
- Computer vision (CV) models are sensitive to noise and cannot generalize patterns
- Poor data quality
- Not enough data

Solutions (Weak performance to train our models)

- Originally CV models ran on CPUs which was not enough to train our complex and large models
- Advancement to GPUs lead to improved calculations for CV models!
- Availability increased and costs decreased

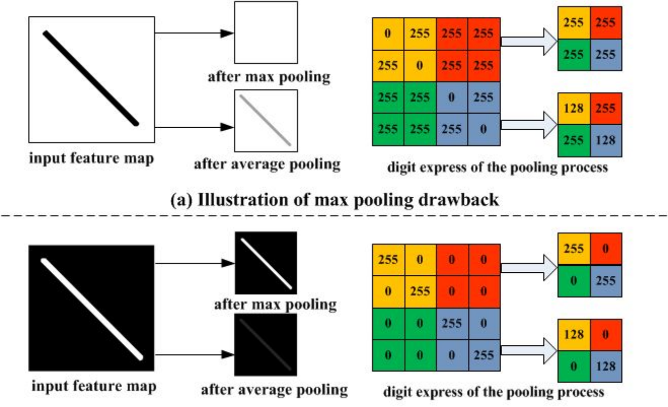






Solutions (Sensitive to Noise/Computation)

- Our models frequently overgeneralize (overfit) and do not learn features
- The images are large → increased computation
  - Solution: Pooling!
  - Each colored box below is a window. We summarize the features in that window by an operation
    - Max pooling (Take the max RGB value in a group of nxn pixels)



0	255	255	255
255	0	255	255
255	255	0	255
255	255	255	0

255	255
255	255

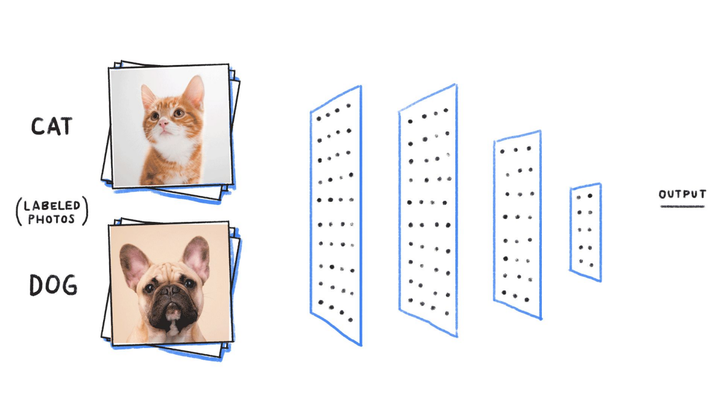
128	255
255	128

(a) Illustration of max pooling drawback

255	0	0	0
0	255	0	0
0	0	255	0
0	0	0	255

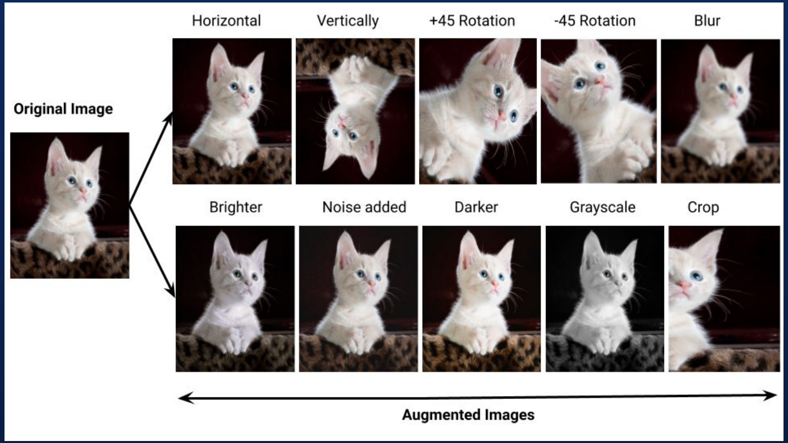
255	0
0	255

128	0
0	128



Solutions (Data Collection)

- A lack of images, results in the model failing to recognize patterns and poor performance
  - Solution: Data Augmentation!
  - We take the original data and modify the image to create new data to feed into the model
    - Methods: rotate, grayscale, color jitter, blur, invert, flip the image (There are more methods!)

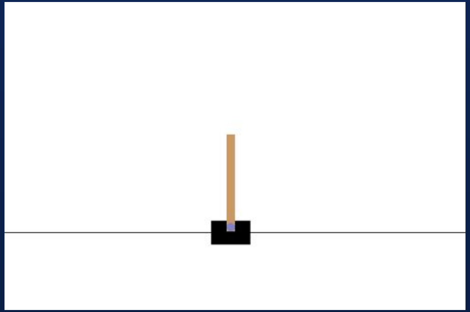


03

RL Example

We will be doing, an Reinforcement learning example!

- Packages
  - random
  - gym
  - tensorflow
  - rl
- Use pip install _ where blank is one of the above package names
- pip install gym==0.25.2 tensorflow==2.12.0 keras-rl2==1.0.5 numpy pyglet pygame protobuf==3.20.*



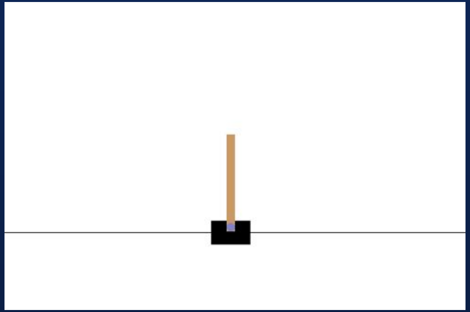
Requirements

- Problem: A pole is attached by an un-actuated joint to a cart, which moves along a frictionless track. The pendulum is placed upright on the cart and the goal is to balance the pole by applying forces in the left and right direction on the cart.

  State: The space with information on the cart pole

  Actions: Push Cart Left, or Push Cart Right

  Agent: The decision-maker that tries to solve the task of balancing the pole



Time to Code!

THANK YOU!

Join BigThink AI!