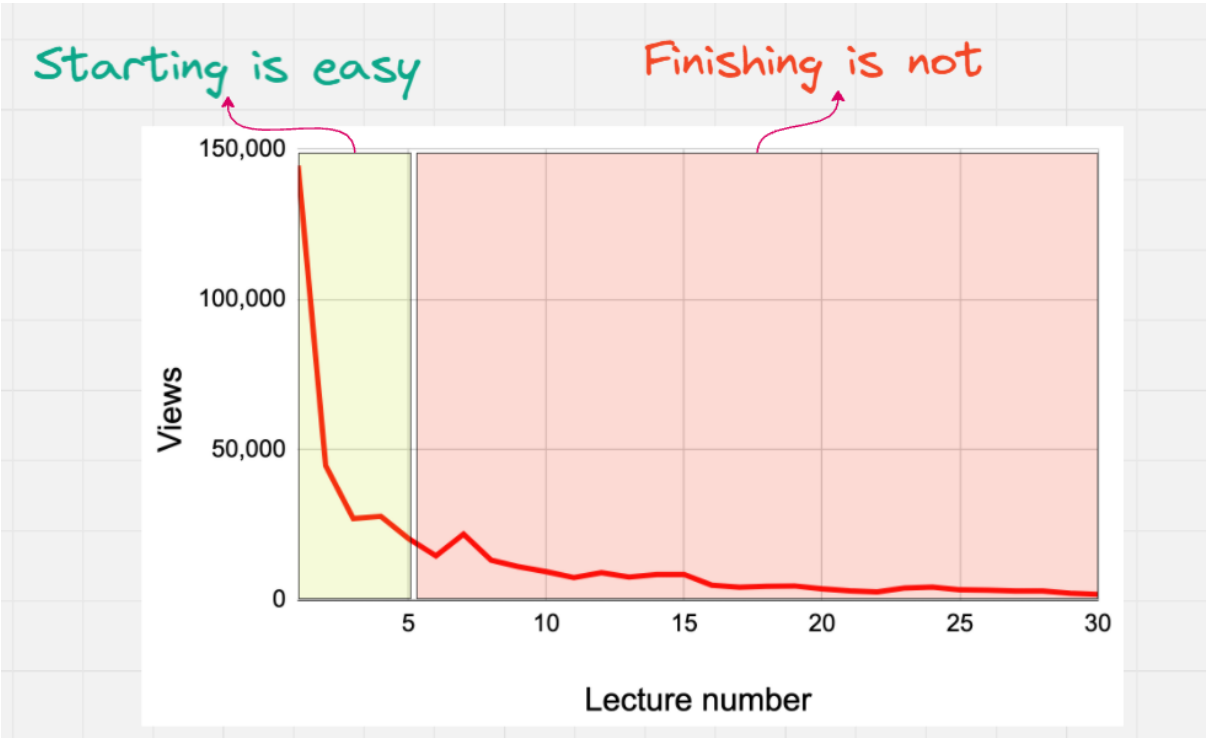
Lecture 1

COMPUTER VISION FROM SCRATCH

Introduction to CV

Before we start



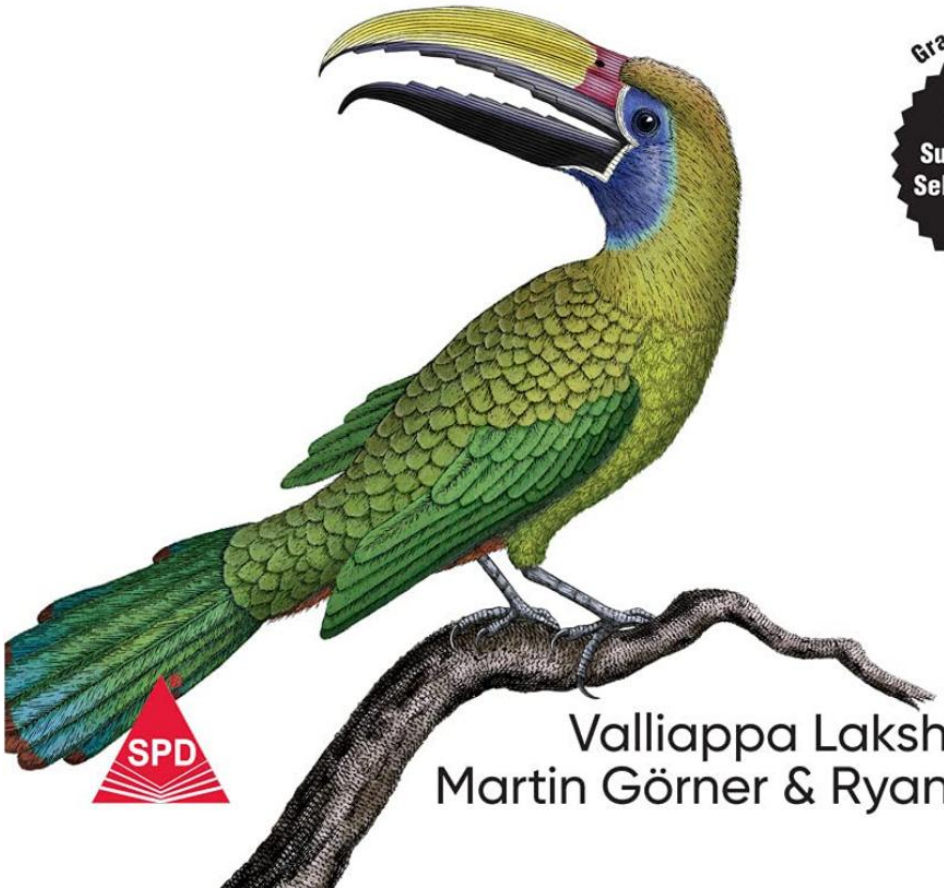
Let us take a pledge to finish what we started

Reference material for this course

O'REILLY®

Practical Machine Learning for Computer Vision

End-to-End Machine Learning for Images



Grayscale Edition

For Sale in the Indian Subcontinent & Select Countries Only*

*Refer Back Cover

SPD

Valliappa Lakshmanan,
Martin Görner & Ryan Gillard

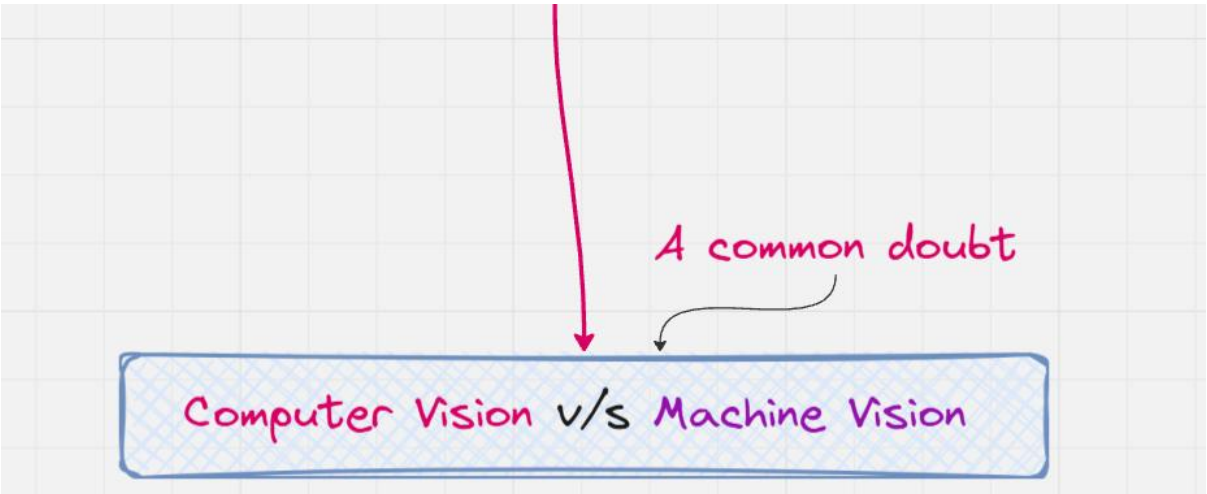
GoogleCloudPlatform/
practical-ml-vision-book

5 Contributors | 6 Issues | 515 Stars | 252 Forks

github.com

GitHub - GoogleCloudPlatform/practical-ml-vision-book

Contribute to GoogleCloudPlatform/practical-ml-vision-book development by creating an account on GitHub.

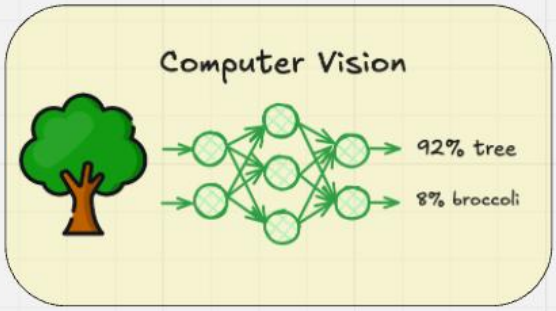


Computer Vision

Enabling machines to understand images/ videos.

It encompasses
- algorithms
- models
- theoretical approaches

Used in
- facial recognition
- self-driving cars
- medical image analysis etc



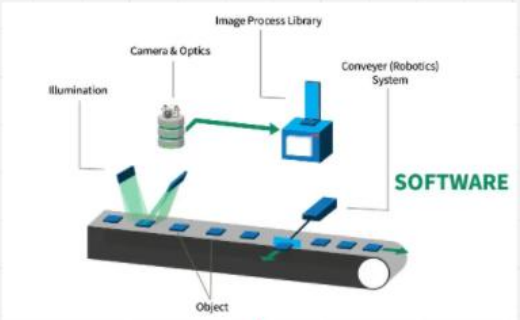
Machine Vision

Practical, industrial use of vision-based systems—often in

- manufacturing
- quality control settings etc

Typically integrate
- cameras
- lighting
- sensors
- specialized software

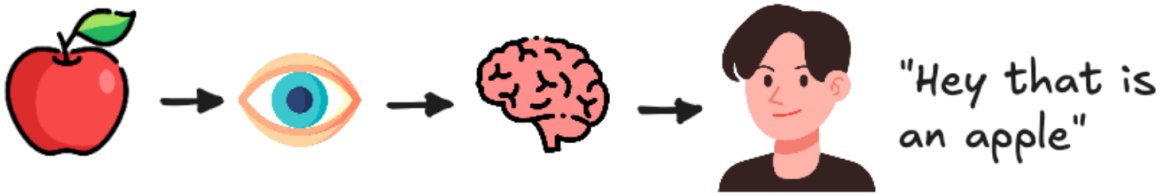
To
- inspect products
- guide robots
- monitor processes in real time etc



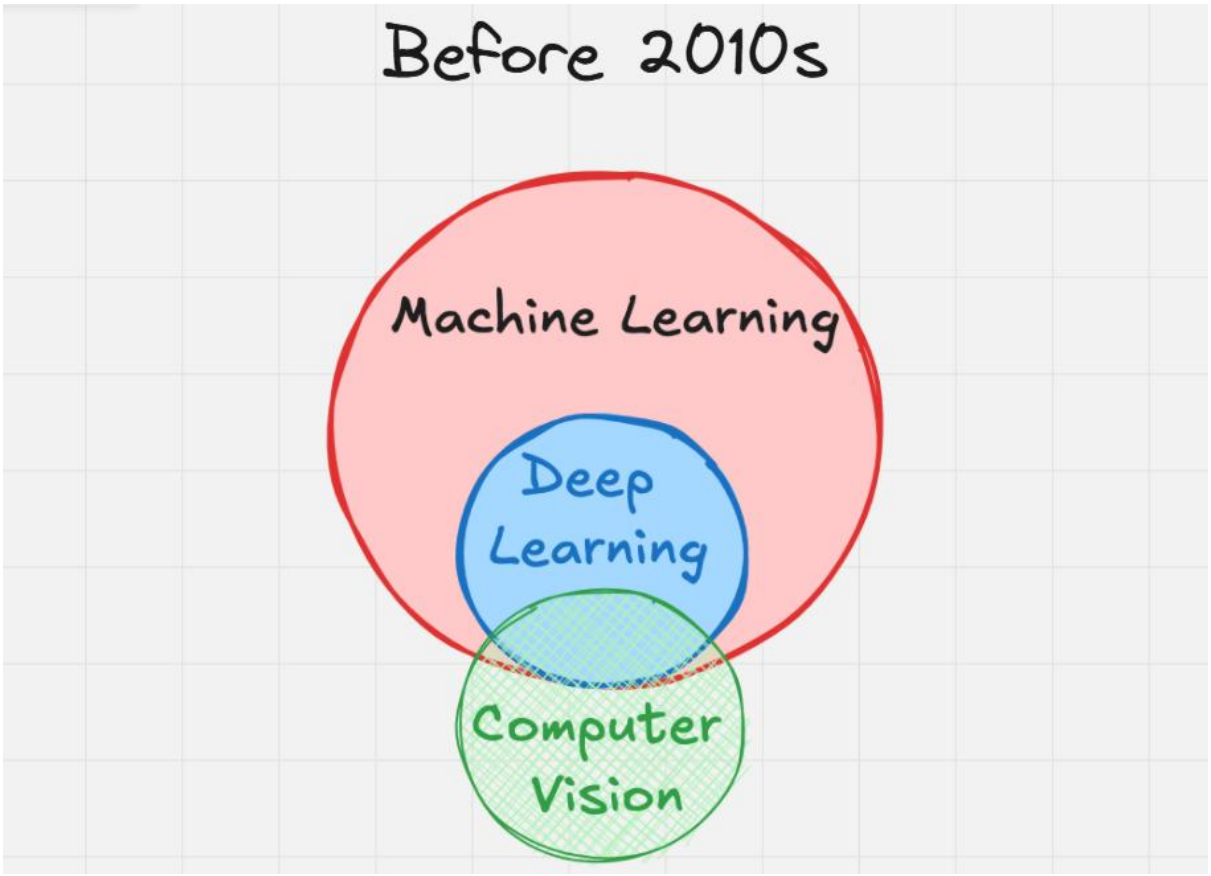
Computer Vision v/s Human Vision

Computer Vision

Human Vision



Computer Vision in the early days

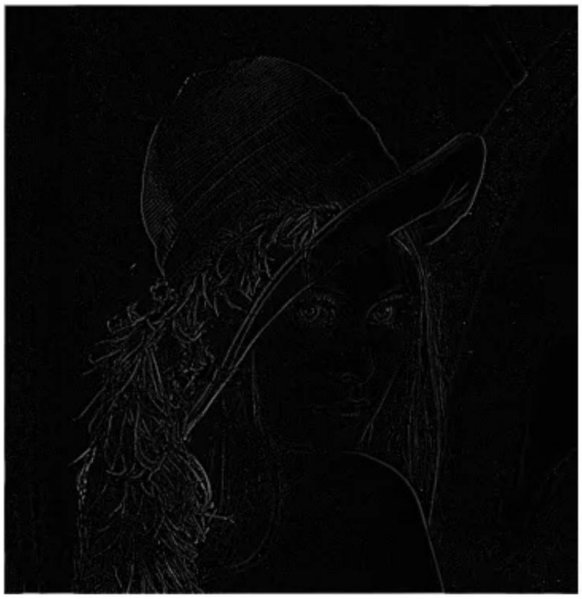


Before 2010, computer vision tasks were performed using image filters. Filters allow you to identify specific features within an image.

Some filters could be meant for edge detection, some could be meant for detecting circular shapes.

Laplacian smoothing filter

Laplace edge detector



Kernel 1

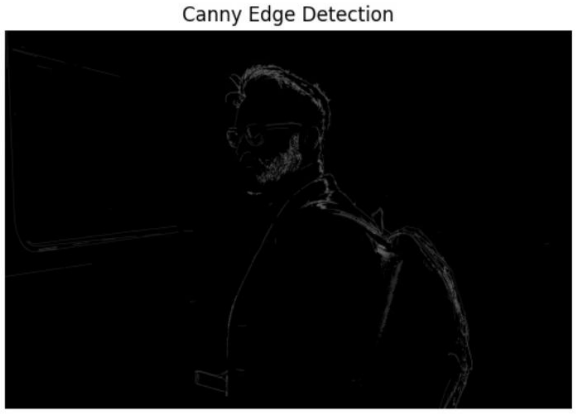
Kernel 2

0	-1	0
-1	4	-1
0	-1	0

Kernel 1

-1	-1	-1
-1	8	-1
-1	-1	-1

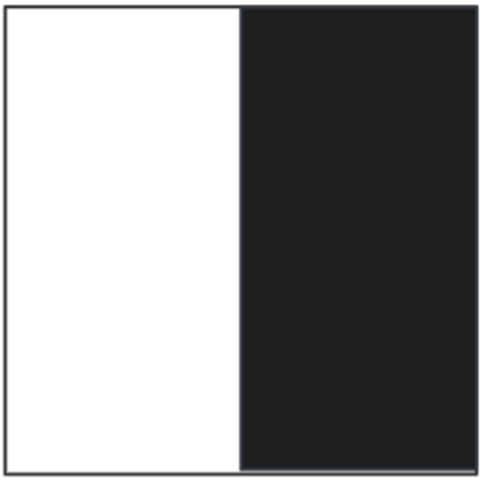
Kernel 2



A quick demonstration of how filters work

Say you want to detect the edge in this image

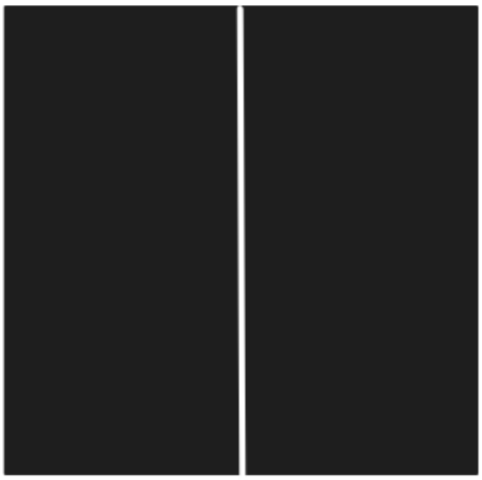
Input image



Pixel values

1	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0

Expected output image



Pixel values

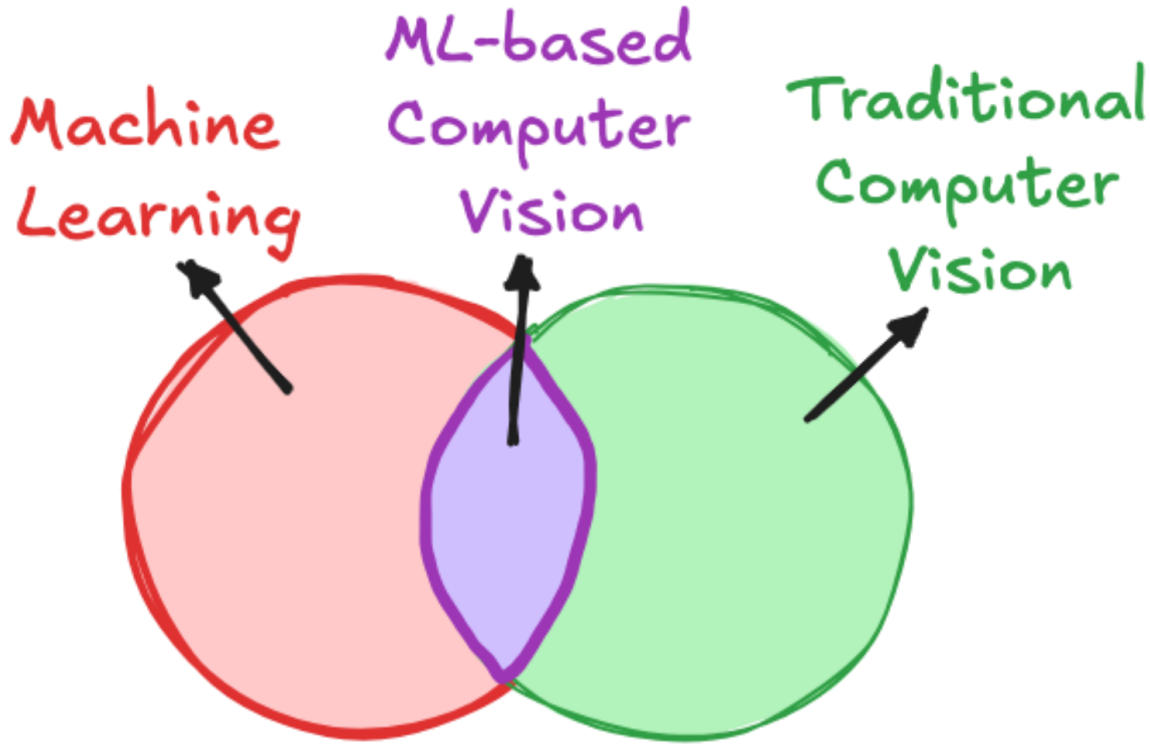
0	0	0	1	1	0	0	0
0	0	0	1	1	0	0	0
0	0	0	1	1	0	0	0
0	0	0	1	1	0	0	0
0	0	0	1	1	0	0	0
0	0	0	1	1	0	0	0
0	0	0	1	1	0	0	0
0	0	0	1	1	0	0	0
0	0	0	1	1	0	0	0
0	0	0	1	1	0	0	0

Computer Vision in the early days: Traditional expert systems

In classical expert-based CV systems, we try to define logic (such as filters) for deciding what is a flower or what is a human from an image. Although good for specific use cases, it was very difficult to make classical expert-based systems work for a multitude of edge cases.

Traditional Rule-based system v/s Machine Learning for CV

In Machine Learning, we let the ML model figure out the logic using large amounts of data. We don't impose any strict logic.



Why is ML based CV good?

If you were to manually define and identify the patterns in an image that identifies a cat (ears, whiskers, eyes etc), then your entire life would be dedicated to creating billions of conditions that can happen in real life images.

1. What if there is a dog and a cat in an image?
2. What if the cat is jumping up a fence?
3. What if the cat is in a snow background?

Just like Ji(a)n Yang from Silicon Valley making hot-dog v/s not-hot dog labels, you will be doing boring scut work.

Think about ChatGPT.

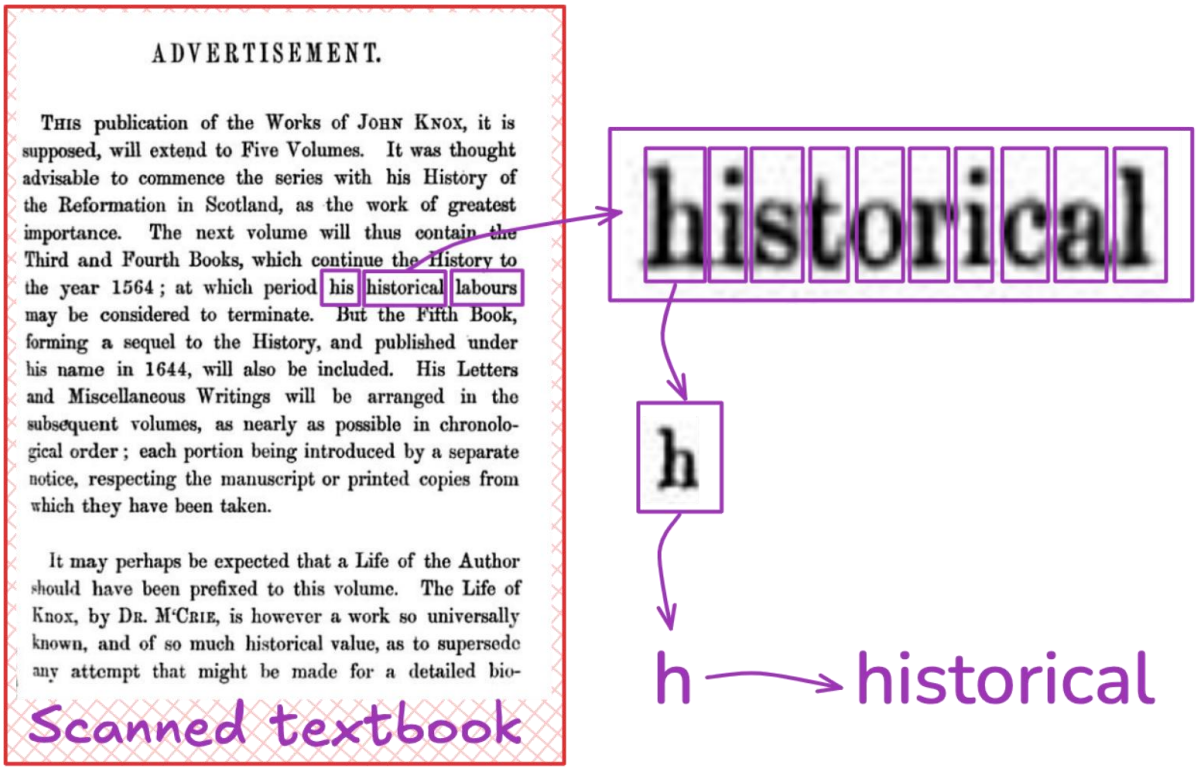
Nobody explicitly gave the LLM instructions to give a pre-determined reply.

LLM figured out how language works by training on the large corpus of internet data.

The only goal was to make next word prediction. Rest of the properties like ability to translate were emergent.

Similarly Machine Learning models can be trained to learn patterns in the image without explicitly telling what to look for. This is what makes the modern Deep Learning based Computer Vision so powerful and useful in applications like Tesla's self driving cars.

Consider the example of Optical Character Recognition (OCR) prior to deep learning



Rule-based (Heuristic) systems

1. Hand-crafted rules: For example, if a certain shape had a loop at the top and a straight line descending on the right side, it might be recognized as "p." These heuristics could get very extensive and required a great deal of manual engineering.

2. Decision trees: Decision trees were sometimes used as a structured way to apply these rules (e.g., "Does the character have a closed loop? Yes/No.").

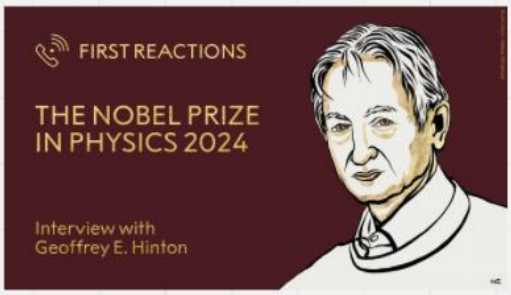
PROBLEMS

1. Variability in fonts and styles
2. Variations in image quality
3. Background complexity
4. Text orientation and alignment
5. Handwritten vs. printed text
6. Language and script diversity
7. Overlapping and connected characters
8. Color and contrast issues
9. Multiline and paragraph structure
10. Special characters, numbers, and symbols

AlexNet: The architecture that changed the history of computer vision in 2012

Image classification was very difficult in the early 2010s due to the usage of expert systems.

However, in 2012 Alex Krizhevsky, Ilya Sutskever (OpenAI co-founder) and Geoffrey E. Hinton (Nobel Laureate in Physics, 2024) published a seminal paper (AlexNet) based on deep convolutional neural networks that changed the way computer vision was implemented.



Till now (Feb 15th, 2025) this paper has been cited more that 170,000 times!

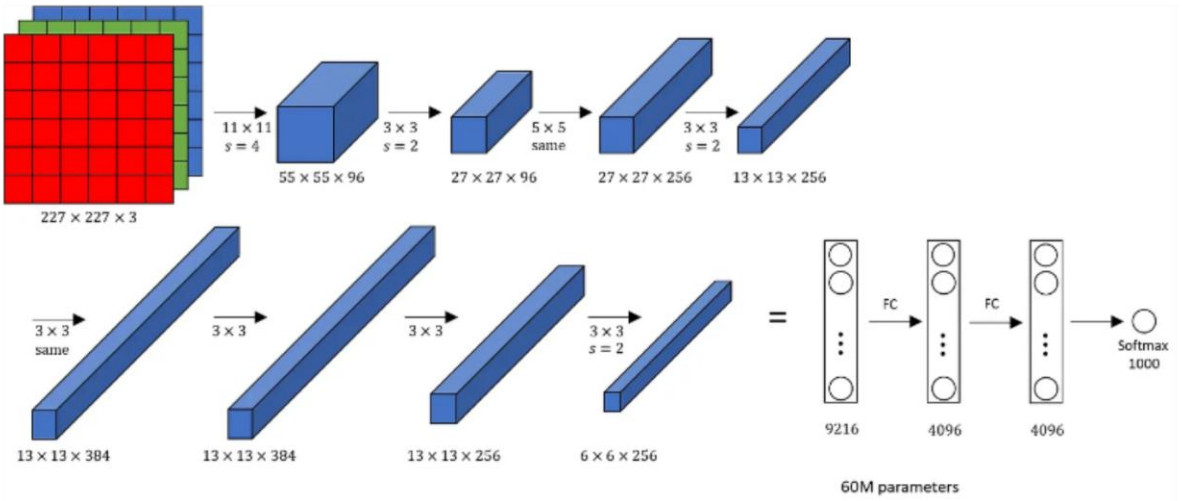
Geoffrey Hinton

FOLLOW

Emeritus Prof. Computer Science, University of Toronto
Verified email at cs.toronto.edu - Homepage

machine learning psychology artificial intelligence cognitive science computer science

TITLE	CITED BY	YEAR
Imagenet classification with deep convolutional neural networks A Krizhevsky, I Sutskever, GE Hinton Advances in neural information processing systems 25	172486 *	2012



What did AlexNet accomplish?

1. Massive performance improvement in ImageNet challenge
   a. AlexNet won the ImageNet challenge 2012 by a huge margin with a top-5 error rate of 15.3% (2nd best model had an error of 26.2%).

2. Deep Neural Networks became mainstream
   a. Before AlexNet, deep learning was mostly just a research topic
   b. AlexNet showed that DL can outperform traditional methods (SVMs, random forests, etc.)

3. Use of GPUs for training
   a. The paper showed that GPUs could dramatically speed up training
   b. Before this, training deep networks was too slow on CPUs

4. Introduced several architectural innovations in the same model
   a. ReLU instead of sigmoid/tanh → Faster training.
   b. Dropout regularization → Reduced overfitting.
   c. Overlapping max pooling → Improved feature extraction.
   d. Data augmentation → Random cropping, mirroring, rotation

Image augme
[random crop
rotation etc

5. Deep Learning took over Computer Vision
   a. Almost overnight, deep learning became the default approach for CV. Tasks like object detection, segmentation, and image generation started being dominated by CNNs.
   b. Later, deep learning expanded into NLP, robotics, healthcare etc.

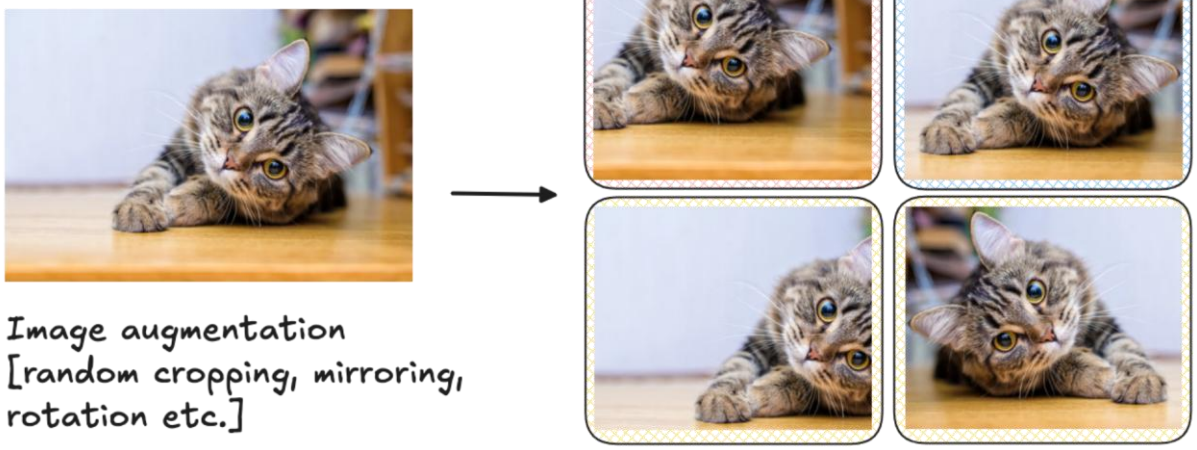
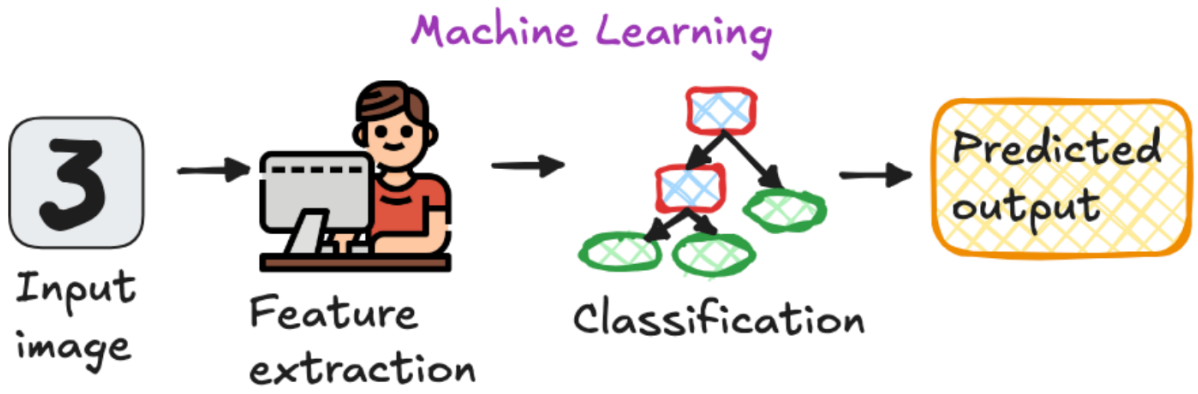


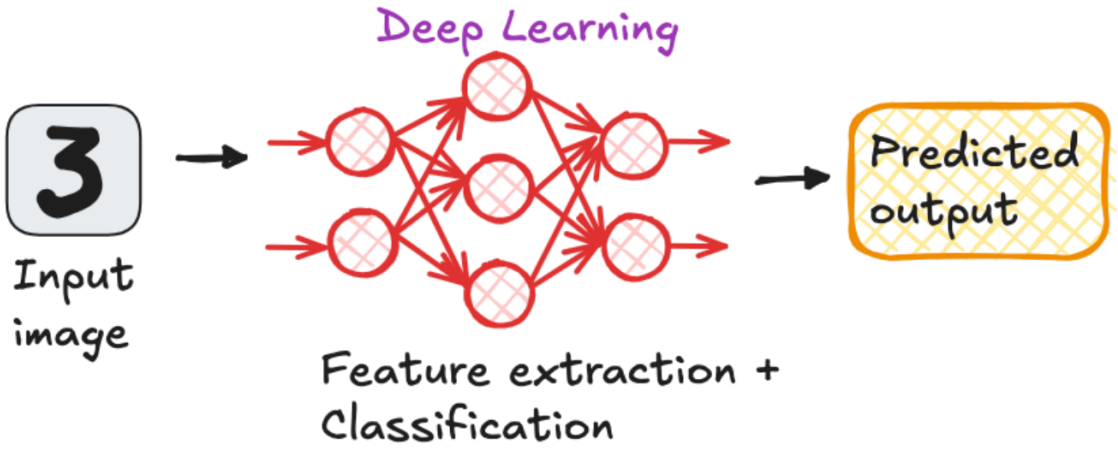
Image augmentation [random cropping, mirroring, rotation etc.]

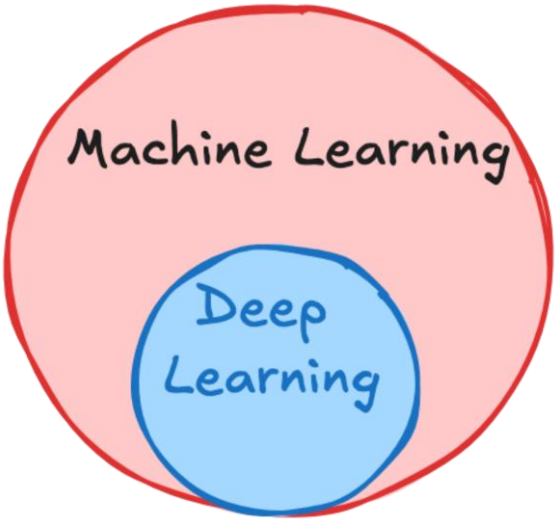
AlexNet's success in ImageNet 2012 triggered the deep learning revolution, proving that deep CNNs could outperform traditional AI approaches, paving the way for the AI boom we see today.

Another common doubt

Machine Learning v/s Deep Learning







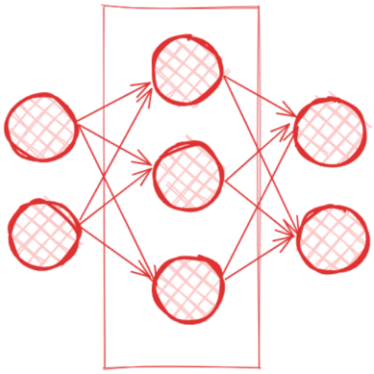
Deep Learning revolution

AlexNet almost single-handedly started the Deep Learning revolution. So what is "deep" in deep learning?

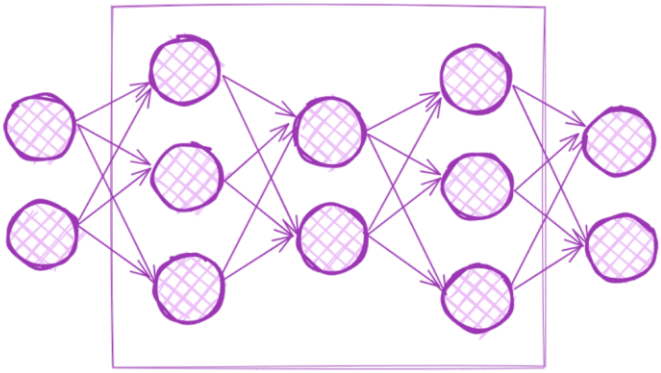
In a shallow neural network you have an input layer, one or two hidden layers, and an output layer.

In a deep neural network you have an input layer, many hidden layers (sometimes dozens or even hundreds), and an output layer. Each layer can learn increasingly abstract features from raw input.

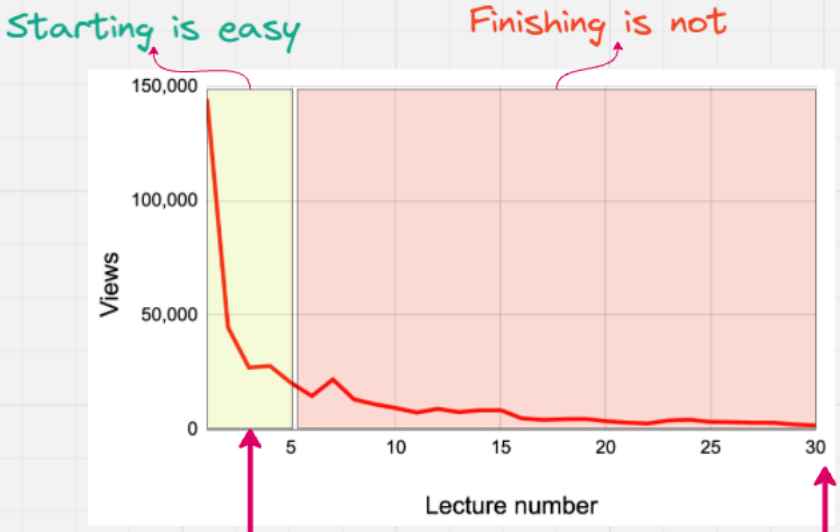
Shallow neural network with 1 hidden layer



Deep neural network with multiple hidden layers



Congrats for finishing a small yet significant step



Be proud of yourself